MASSACHUSETTS INSTITUTE OF TECHNOLOGY

ON THE DETERMINATION AND OPTIMIZATION OF NOMINAL TRAJECTORIES FOR THE REENTRY PHASE OF THE SPACE TRANSPORTATION SYSTEM (STS)

by
Paul William Chin, Jr.

August 1970

Degree of Master of Science

(NASA-CR-141476) ON THE DETERMINATION AND OPTIMIZATION OF NOMINAL TRAJECTORIES FOR THE REENTRY PHASE OF THE SPACE TRANSPORTATION SYSTEM (STS) M.S. Thesis (Draper (Charles Stark) Lab., Inc.) 241 p

N75-70810

Unclas
00/98 08650

T-643

PREPARED AT

CHARLES STARK DRAPER LABORATORY
CAMBRIDGE, MASSACHUSETTS, 02139

ON THE DETERMINATION AND OPTIMIZATION OF NOMINAL TRAJECTORIES FOR THE REENTRY PHASE OF THE SPACE TRANSPORTATION SYSTEM (STS)

by

PAUL WILLIAM CHIN, JR.

S.B., Massachusetts Institute of Technology

(1969)

SUBMITTED IN PARTIAL FULFILLMENT
OF THE REQUIREMENTS FOR THE
DEGREE OF MASTER OF SCIENCE

at the

MASSACHUSETTS INSTITUTE OF TECHNOLOGY

August, 1970

Signature of Author ______________________
Department of Aeronautics and Astronautics, August 1970

Certified by ______________________
Thesis Supervisor

Accepted by ______________________
Chairman, Departmental Graduate Committe

ON THE DETERMINATION AND OPTIMIZATION OF NOMINAL TRAJECTORIES FOR THE REENTRY PHASE OF THE SPACE TRANSPORTATION SYSTEM (STS)

by

Paul W. Chin, Jr.

Submitted to the Department of Aeronautics and Astronautics in August, 1970 in partial fulfillment of the requirements for the degree of Master of Science.

ABSTRACT

The purpose of this thesis is to determine and optimize nominal reentry trajectories for the orbiting stage of the Space Transportation System (STS) employing a new differential dynamic programming approach. The trajectories are primarily based on the motion of the center of mass of the vehicle configuration with secondary attention given to stability and control which is concerned with the vehicle attitude motions about this center of mass.

The STS reentry is divided into three phases: the hypersonic reentry; the transition to cruise; and the powered flight cruise, approach, and landing. Principal attention is directed at the first two phases in an integrated manner. The Apollo reentry system serves as a guide to the hypersonic phase analysis. The difference introduces itself in the form of a transonic transition to cruise, where the hypersonic, linear assumptions no longer remain valid. In this low Mach number region, non-linearities and higher order terms may not be neglected, and must be reinstated into the dynamic model and the equations of motion. Aerodynamic control is employed in the form of a lift vector rolling essentially about the wind axis or velocity vector. Fundamental constraints include a maximum acceleration, total heat accumulation and heat rate limitations, and the terminal accuracy desired for the start of the powered flight cruise.

New second-order and first-order differential dynamic programming techniques [1-2] for determining optimal control with wider application than existing second-variation and second-order methods are applied. Advantages of such an approach include: lessening the restriction of maintaining a globally positive definite inverse second partial derivative matrix of the Hamiltonian with respect to the control vector, H_{uu}, so that a larger class of problems may be handled adequately; rapid convergence of non-linear problems; and stability of integrations along nominal non-optimal trajectories.

The object is to apply this new technique to the STS reentry control problem. The determination and selection of nominal trajectories are influenced more by optimization procedures than by operational considerations. But more important than optimality is the methodology of design which optimal control provides. Much attention can then be devoted to the developing trends, perhaps with respect to some nominal reference. The changeability of the present NASA STS strawman configuration coupled with the sparsity of aerodynamic data in the transonic region makes such a generalized approach very feasible and desirable. With a prudent choice of state and control variables, the sensitivity parameters should reveal themselves, leading to the desired guidance law. The emphasis is to show the validity and worthwhileness of such a flexible approach in determining optimal reentry trajectories.

ACKNOWLEDGEMENTS

I wish to express my appreciation to my thesis advisor, Prof. John J. Deyst, Jr., for supervising this thesis and for his constructive criticism and helpful suggestions throughout this entire effort.

The advice, comments, and suggestions of my associates at the M. I. T. Draper Laboratory was extremely helpful and greatly appreciated. In particular I would like to thank Mr. Chung Pu for numerous and valuable technical discussions on the subject of differential dynamic programming.

The cooperation of Mrs. Susan MacDougall, Mrs. Beth Hwoschinsky, and Mrs. Carole Taylor, who typed the drafts and final manuscript, the help of Mr. William Eng, who prepared the figures, and the assistance, perseverance, and patience of the staff of the Digital Computation Group are appreciated.

This report was prepared under DSR Project 55 40800, sponsored by the Manned Spacecraft Center of the National Aeronautics and Space Administration through Contract NAS 9-10268 with the Charles Stark Draper Laboratory of the Massachusetts Institute of Technology in Cambridge, Massachusetts.

The publication of this report does not constitute approval by the M. I. T. Draper Laboratory nor the National Aeronautics and Space Administration of the findings or the conclusions contained herein. It is published only for the exchange and stimulation of pertinent and useful ideas.

TABLE OF CONTENTS

FIGURES

NOTATION

a	Difference between optimal cost $J^{o}[\bar{\underline{x}}(t), t]$ using $\underline{u}^{o}(\tau) = \bar{\underline{u}}(\tau) + \delta\underline{u}^{o}(\tau)$ and nominal cost $\bar{J}\,[\bar{\underline{x}}(t), t]$ using $\bar{\underline{u}}(\tau)$, $\tau \epsilon [t, t_f]$
c	Critical value for $\Delta J \,/\, \lvert a[\bar{\underline{x}}(t_1), t_1] \rvert$ in nearness test
C_D	Aerodynamic drag coefficient
C_L	Vertical trajectory plane component of aerodynamic lift coefficient
$C_{L\,max}$	Aerodynamic lift coefficient
d	Reference distance from vehicle nose to critical stagnation points in ft.
D	Drage force in lbs.
dt	Time step duration in sec.
$\underline{f}$	Time derivative of the state vector equivalent on right hand side of state equation
F	Terminal cost or performance index
g	Earth gravitational acceleration equal to 32.2 ft/sec.^{2}
g's	Vehicle acceleration, in units of g

GF_1	Vehicle configuration factor
GF_2	Vehicle wing configuration geometry factor
GF_3	Vehicle wing configuration geometry factor
h	Local vertical earth altitude in ft.
H	Hamiltonian
h_s	Atmospheric scale height utilized in calculating the exponentially varying atmospheric density, in ft.
J	Total cost function or performance index
L	Accrued cost function or performance index
L	Vertical trajectory plane component of lift force in lbs.
L_{max}	Lift force in lbs.
L/D	Vertical trajectory plane component of lift-to-drag ratio
L/D, $L/D]_{max}$	Lift-to-drag ratios
m	Vehicle mass in slugs
M	Mach number
q	Dynamic pressure in slugs/ft/sec^2
q_c	Total accumulated convective heating in BTU/ft^2
$\dot{q}_c$	Convective heat rate in BTU/ft^2/sec.

r	The $r^{\underline{th}}$ value of t_1 encountered in attempting to satisfy the nearness criteria
R	Magnitude of earth centered position vector equivalent to sum of local vertical altitude and some representative Earth radius in ft.
R_e	Representative Earth radius in ft.
R_N	Representative vehicle nose or leading edge radius in ft.
S	Vehicle reference wing area
t	Time in sec.
t_b	Time of instability in matrix Riccati reverse differential equation, in sec.
t_{eff}	Time at which the trajectory first violates a necessary condition of optimality, or when $\|a\|$ first exceeds η, in sec.
t_f	Terminal time in sec.
t_o	Initial time in sec.
$t_{o, r}$	$r^{\underline{th}}$ value of t_1 encountered in attempting to satisfy the nearness criteria, in sec.
t_1	Time at which nominal control is replaced by new optimal control, or the earliest time when the nearness criteria is satisfied, in sec.

t_2	Beginning of time interval where the nominal trajectory is the optimal trajectory over $[t_2, t_f]$, but not optimal over $[t_o, t_f]$
$\underline{u}$	Control vector
v	Relative velocity in ft/sec
v_∞	Free stream velocity
W	Vehicle weight in lbs.
$W_{(\ldots)}$	Quadratic penalty weighting factor for subscripted quantity
$\underline{x}$	State vector
x_r	Down range distance on Earth surface in ft. or n. m.
α	Vehicle angle of attack in radians
β	Optimal linear feedback controller maintaining the necessary condition of optimality
γ	Vehicle flight path angle positive upward, in radians
η	Small positive tolerance quantity for $\lvert a \rvert$ in determining a suitable t_1
ρ	Local Earth atmospheric density in slugs/ft^3
ρ_o	Local Earth atmospheric density at sea level in slugs/ft^3
σ	Standard deviation of a distribution

τ Dummy variable for time in sec.

φ Vehicle roll angle in radians

ψ Terminal boundary condition

$\sigma^2_{(\ldots)}$ Denotes an acceptable tolerance or variance in the subscripted quantity

Δt A small time interval in sec.

$\underline{(\ldots)}$ Under score denotes a column vector

$\overline{(\ldots)}$ Overbar denotes a nominal quantity

$\dot{(\ldots)}$ Dot notation indicates total derivative with respect to time

$(\ldots)^o$ Superscript o denotes optimal

$(\ldots)^*$ Superscript * denotes determination of the quantity by satisfying the necessary condition of optimality

$\min_{(\ldots)} \{\ldots(\ldots)\ldots\}$ Denotes minimization of quantity in braces with respect to parenthesized quantity

$(\ldots)^T$ Denotes matrix or vector transposed

$[\ ]^{-1}$ Denotes matrix inverse

$|(\ldots)|$ Denotes absolute value of quantity

$\delta(\ldots)$ Denotes a deviation in the quantity from its reference value

$\delta(\ldots)$ Denotes delta penalty terms

$\tilde{\delta}(\ldots)$	Denotes delta penalty terms
$\Delta(\ldots)$	Denotes a quantity difference
$(\ldots)]_{max}$	Denotes a maximum desired quantity threshold value for cost penalization
$(\ldots)]_{desired}$	Denotes desired quantity range threshold values for cost penalization
$\{\ldots\}]_{(\ldots)}$	Denotes the quantity in braces evaluated at the subscripted value
$(\ldots)_0$	Subscript 0 denotes an initial time quantity
$(\ldots)_f$	Subscript f denotes a terminal time quantity
$(\ldots)_s$	Denotes partial differentiation with respect to a scalar s
$(\ldots)_{\underline{s}}$	Denotes partial differentiation with respect to a vector $\underline{s}$ to yield a row vector
$(\underline{\ldots})_{\underline{s}}$	Denotes partial differentiation of a column vector with respect to a vector $\underline{s}$ to yield a matrix of dimension: (dimension of parenthesized vector) x (dimension of $\underline{s}$)
$(\ldots)_{\underline{r}\,\underline{s}}$	Denotes partial differentiation with respect to column vectors $\underline{r}$ and $\underline{s}$ to yield a matrix of dimension: (dimension of $\underline{r}$) x (dimension of $\underline{s}$)

CHAPTER I

INTRODUCTION

In recent years, much attention has been devoted to the problem of guiding a lifting-body reentry vehicle through an earth atmospheric reentry to a desired target safely. This had to be accomplished subject to the total heat accumulation and heat rate limitations of the spacecraft shielding material, and the maximum acceleration and load factor limitations of the crew and vehicle structure. It was also highly desirable to be able to direct the spacecraft to within a certain radius of a specified target.

These constraints have been examined both individually and collectively. For a vehicle reentering the earth's atmosphere at a given velocity, there is a certain amount of peak heating rate and maximum acceleration that is intrinsic to a system possessing such amounts of kinetic energy. Parametric studies can only attempt to reduce these by changing their altitude, time, and to some extent duration of occurence. It seemed desirable to reduce the peak heating rate by maintaining a higher altitude where the atmosphere is less dense, but however prolonging the portion of the entry and increasing the total amount of heat accumulated by the vehicle. It was found to be far more profitable to descend lower, incurring higher peak heating rates, but decreasing the duration and thus the total heat accumulated. Maximum load factor and acceleration patterned those of the peak heating rates quite closely. This again is due to the kinetic energy of the vehicle.

Aerodynamic control of the lifting-body reentry vehicle is employed by proper directioning of the lift vector. For ballistic reentry, experience with the Mercury spacecraft indicated a state-of-the-art landing point accuracy of twenty to forty nautical miles (1σ). For entry using low L/D (less than 0.5), experience with the Gemini spacecraft indicated a state-of-the-art landing point accuracy of approximately four nautical miles (1σ), where the navigation error is about three nautical miles (1σ)[N-1]. Results of the Apollo flights to date indicate even smaller miss distances[M-5], with the Apollo 12 flight missing by about one nautical mile[C-3]. The L/D for Apollo was about 0.3, with the maximum amount of lift limited by the vehicle trim angle of attack.

In the case of the STS orbiter or shuttle vehicle, the maximum capable amount of lift is to be determined by the angle of attack history. The vehicle configuration is similar to a conventional airplane, affording a potential for relatively low development costs. The basic shuttle concept is depicted in Fig. 1.1, taken from the Manned Spacecraft Center (MSC) Shuttle Status briefings[M-1]. Both the booster and orbiter stages employ airplane type practices. The primary concern of this thesis is with the reentry of the orbiter stage.

Because of the critical transonic transition to cruise maneuver, the hypersonic linear analysis is no longer valid for the whole duration of the orbiter reentry as it was in the Apollo reentries. Non-linearities and higher order terms can no longer be neglected. For this now more complex situation, the problem of determining optimal nominal reference trajectories demands a more efficient optimization procedure. The lifting-body atmospheric reentry formulation is already quite complex, even with the linear hypotheses. Coupled with the fact that much of the vehicle aerodynamic characteristics are unknown due to the non-existence of a true test model, and non-linear as well, the problem is further complicated.

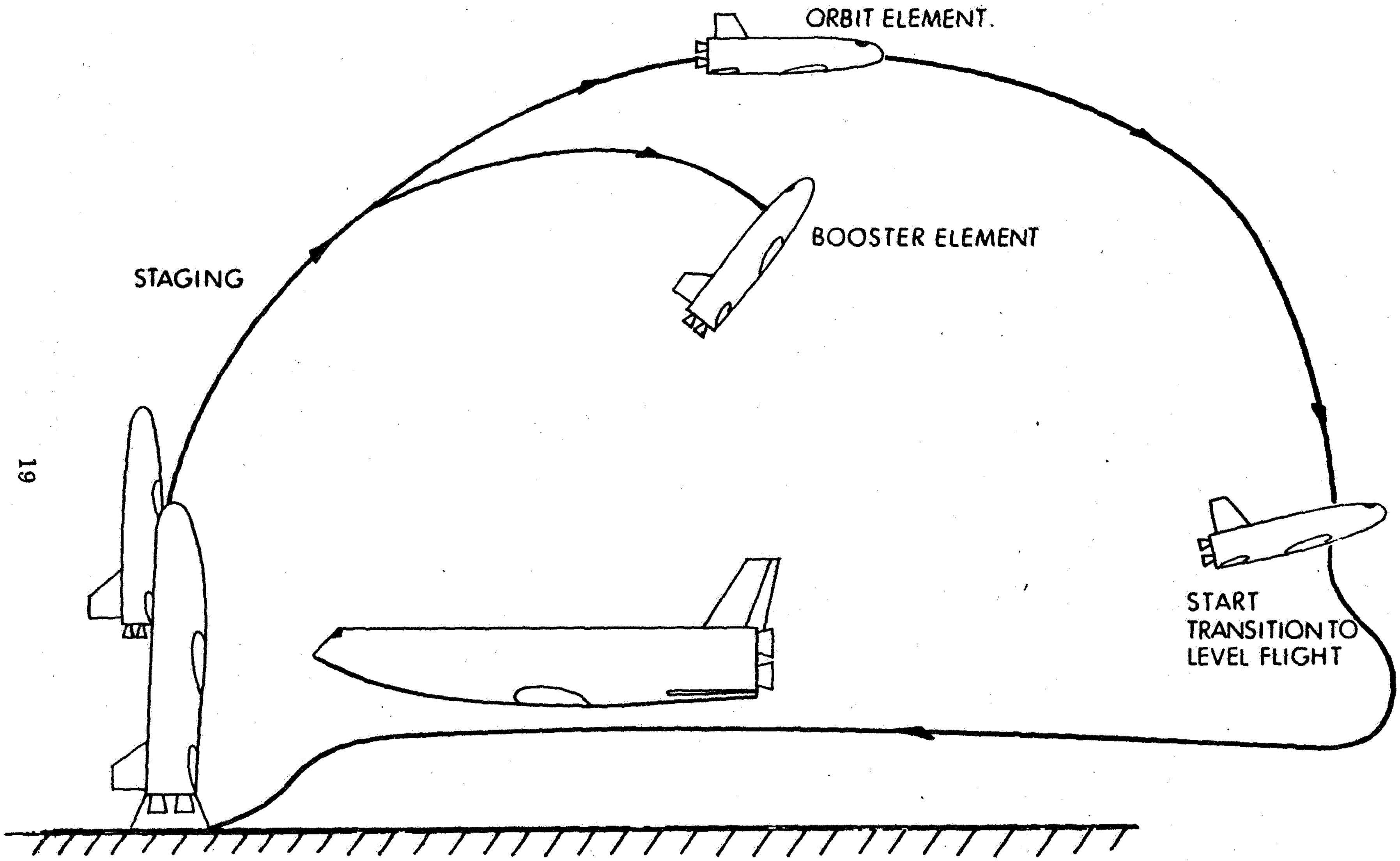


Figure 1.1 STS Shuttle Concept

The utilization of a second-order or second-variation functional optimization technique seems the logical solution, and there have been several of these techniques which have been applied to the non-linear reentry problem. However, there arises the question as to the worthwhileness of such an approach, basically from an efficiency point of view. The non-linear reentry problem was met with varying degrees of success, and problems arose with the instability of integrations, the rate of convergence to the desired solution in terms of the number of iterations and the computation time required, and even the applicability of certain of these techniques. There emerged, from the notion of differential dynamic programming, new second-order and first-order techniques for determining optimal control with wider application than existing second-order and second-variation methods[J-2]. Using this method, the fact that the inverse second partial derivative matrix of the Hamiltonian with respect to the control vector, H_{uu}, was not required to be globally positive definite allowed a larger class of problems to be considered. The new step-size adjustment routine facilitated the rapid convergence of non-linear problems and ensured the stability of the backward integration along nominal non-optimal trajectories. The variables of the backward integration are themselves the first and second order influence functions, and lead to the desired perturbation guidance law.

The object is to show the applicability of this new approach to the non-linear STS reentry problem. The main concern is more with determining trends and influence factors than with solving for the optimal solution, which optimal solution depends on the form of the cost functional or performance index and the weighting of the state and control parameters, which in turn depend upon the information sought. Numerical studies further the understanding of maneuvers, but the number and variety of boundary conditions and constraints preclude a very effective use of an experimental "open loop" approach to optimal nominal trajectory determination. Optimal control techniques reduce the guess work affording a consistent framework within which to generate and evaluate control strategies [S-1].

CHAPTER II

THE REENTRY FORMULATION

2.1 Reentry

The atmospheric reentry analysis is primarily concerned with the motion of the center of mass of a lifting-body vehicle. The attitude motions of the vehicle about its center of mass are usually not incorporated directly into the formulation, but instead introduced through penalty functions or other means. The reentry phase begins at an altitude of 400,000 feet above the surface of the Earth, although the altitude at which the aerodynamic forces become significant, depending on the velocity and geometric characteristics of the vehicle, occurs at about 300,000 feet above the Earth's surface.

For the purposes of trajectory determination, the Earth is modelled as a non-rotating homogeneous sphere, surrounded by an exponentially varying dense atmosphere. The only aerodynamic forces taken into account are the lift and drag of the vehicle. The other forces considered are due to the gravitational attraction of the Earth and the centripetal acceleration of the vehicle.

For the heating model, only convective heating is taken into account. Radiative heating may be neglected because the vehicle velocity is less than 30,000 feet per second. The critical heating locations for such a vehicle configuration are at the nose stagnation point, the wing leading edge, the aft wing region, and the three dimensional apex area behind the nose stagnation region, as illustrated in Fig. 2.1.1. The skin temperature at any point of these regions can be calculated by means of heat-balance equations.

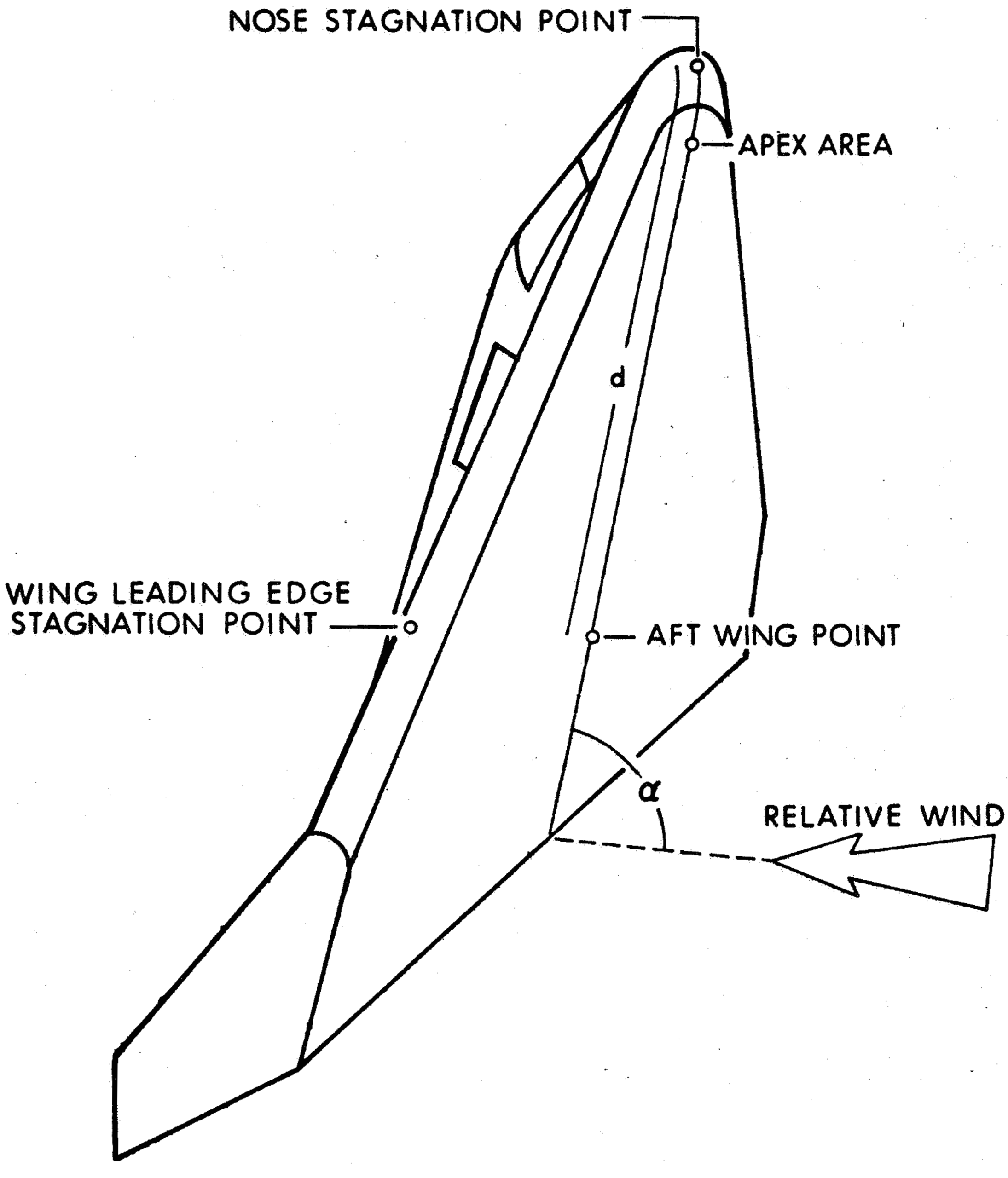


Figure 2.1.1 Critical Heating Locations on a Reentry Vehicle

2.2 Equations of Motion

Consider the two-dimensional trajectory planar motion of a lifting-body reentry vehicle as depicted in Fig. 2.2.1. The dynamic equations of motion of the center of mass of the vehicle system configuration along and orthogonal to the velocity vector in local earth coordinates are:

$$m \frac{dv}{dt} = - D - mg \sin\gamma + m \frac{v^2}{R} \sin \gamma \qquad (2.2.1)$$

$$m \frac{d}{dt}(v\gamma) = L - mg \cos \gamma + m \frac{v^2}{R} \cos \gamma \qquad (2.2.2)$$

The remaining equations of motion are the kinematic relationships describing the motion of the vehicle center of mass:

$$m \frac{dx_r}{dt} = mv \cos \gamma \frac{R_e}{R} \qquad (2.2.3)$$

$$m \frac{dh}{dt} = mv \sin \gamma \qquad (2.2.4)$$

Assuming that the velocity is essentially constant over the duration of an integration time step, and introducing the following aerodynamic relationships,

$$D = \frac{1}{2} \rho v^2 S C_D \qquad (2.2.5)$$

$$L = \frac{1}{2} \rho v^2 S C_L \qquad (2.2.6)$$

where L represents the trajectory plane vertical component of the lift force, along with the relationship:

$$R = R_e + h \qquad (2.2.7)$$

we rewrite the equations of motion in the particular form:

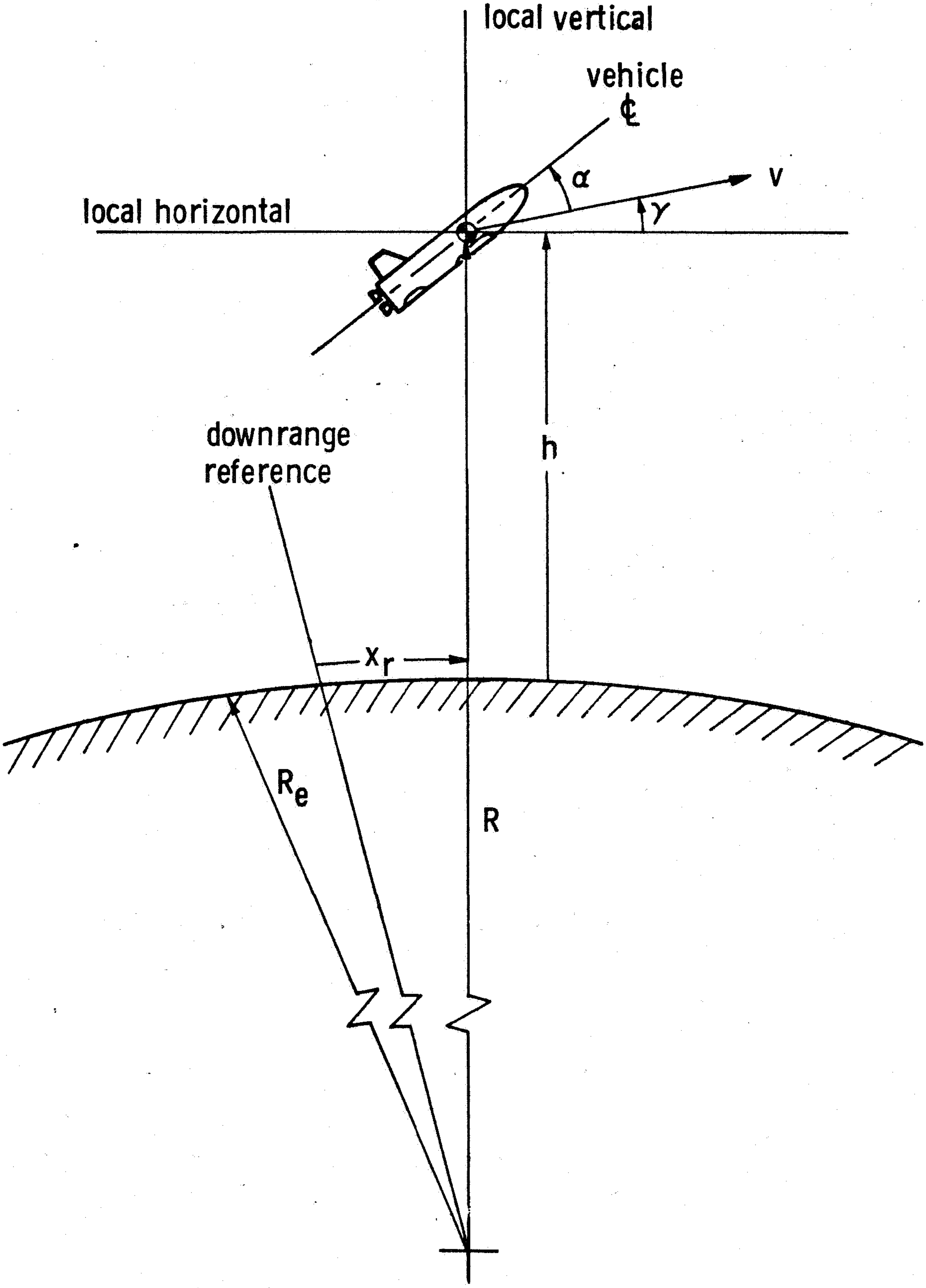


Figure 2. 2. 1 Vehicle Trajectory Coordinate Geometry

$$\dot{x}_r = \frac{v \cos \gamma}{(1 + h/R_e)} \qquad (2.2.8)$$

$$\dot{v} = -\frac{1}{2} \rho v^2 \frac{C_D g}{W/S} - g \sin \gamma \frac{v^2}{(R_e+h)} \sin \gamma \qquad (2.2.9)$$

$$\dot{\gamma} = \frac{1}{2} \rho v \frac{C_L g}{W/S} - \frac{g}{v} \cos \gamma + \frac{v}{(R_e+h)} \cos \gamma \qquad (2.2.10)$$

$$\dot{h} = v \sin \gamma \qquad (2.2.11)$$

Equations (2.2.8) - (2.2.11) are the basic equations of motion for the lifting-body reentry problem in local coordinates, where x_r represents the downrange, v is the relative velocity, γ the flight path angle measured positive upward from the local horizontal, and h the local vertical altitude. S is the representative vehicle wing area, ρ the local density of the atmosphere, and R_e some representative radius of the Earth.

There are many ways of representing these equations of motion, each with its particular advantages and peculiarities. The geometry of the situation suggests that spherical or polar coordinates are the more natural. Chapman[C-1] extends this concept with the introduction of a special Z transformation variable, essentially to reduce the dynamic equations of motion to a single expression. From a purely mathematical point of view, this is aesthetically pleasing, and the computations are considerably simplified by the use of acceleration (density-altitude) inputs. However, the approximations assumed in that formulation are contrary to this author's intentions to include all the important terms in the dynamic equations of motion for this particular non-linear atmospheric reentry problem formulation. Also, the form of Eqs. (2.2.8)-(2.2.11) is simpler to understand, and the physical significance of each quantity is much more readily identifiable. Here, mathematical and computational efficiency have been sacrificed in favor of comprehensibility.

The convective heat rate equations or heat flux equations for the four critical heating areas are:

$$\dot{q}_C = \frac{17,600}{\sqrt{R_N}} \left(\frac{\rho}{\rho_0}\right)^{0.5} \left(\frac{v}{26,000}\right)^{3.15} \tag{2.2.12}$$

for the nose stagnation region,

$$\dot{q}_C = \frac{8,800\sqrt{3}}{\sqrt{d}} \left(\frac{\rho}{\rho_0}\right)^{0.5} \left(\frac{v}{26,000}\right)^{3.15} GF_1 \tag{2.2.13}$$

for the apex region,

$$\dot{q}_C = \frac{12,400}{\sqrt{R_N}} \left(\frac{\rho}{\rho_0}\right)^{0.5} \left(\frac{v}{26,000}\right)^{3.15} GF_2 \tag{2.2.14}$$

for the wing leading edge, and

$$\dot{q}_C = \frac{8,800}{\sqrt{d}} \left(\frac{\rho}{\rho_0}\right)^{0.5} \left(\frac{v}{26,000}\right)^{3.15} GF_3 \tag{2.2.15}$$

for the aft wing region, where d is the distance from the nose, and GF_1, GF_2, and GF_3 are a vehicle configuration factor and geometry factors for the particular vehicle wing configuration. R_N is the representative nose or leading edge radius, while ρ_0 is the sea level atmospheric density.

The total amount of convective heat accumulated is simply the integral of the convective heat rate over the total time interval of the reentry:

$$q_C = \int_{t_0}^{t_f} \dot{q}_C \, dt \tag{2.2.16}$$

The acceleration in g's assumes an especially simple form:

$$g's = \frac{1/m}{32.2} \sqrt{L_{max}^2 + D^2} \tag{2.2.17}$$

or in another more useful form;

$$g's = \frac{q}{W/S} \sqrt{C_D^{\,2} + C_{L_{max}}^{\,2}} \tag{2.2.18}$$

where q is the dynamic pressure represented by:

$$q = \frac{1}{2} \rho v^2 \tag{2.2.19}$$

$C_{L_{max}}$ and L_{max} are the maximum coefficient of lift and maximum lift force for a particular angle of attack, respectively. These are related to their trajectory plane vertical components by the following expressions:

$$C_L = C_{L_{max}} \cos \phi \tag{2.2.20}$$

$$L = L_{max} \cos \phi \tag{2.2.21}$$

Here, ϕ is the roll angle or bank angle about the velocity vector. A perhaps more illuminating expression is of the form:

$$L/D = (L/D)_{max} \cos \phi \tag{2.2.22}$$

where $(L/D)_{max}$ is determined by the angle of attack, α.

2.3 The Apollo Hypersonic Reentry Concept

Selecting the proven[M-5] Apollo reentry concept as a starting point, we recall that for its equations of motion, Eqs. (2.2.8) - (2.2.11) were employed assuming a small flight path angle, γ, and neglecting the component of centripetal acceleration along the velocity vector[L-4, M-4]. In the case of the Apollo earth orbital reentries, the initial velocity at entry interface (h = 400,000 ft.) was close to 26,000 fps. Thus the portion of the Apollo reentry that pertains to the STS orbital reentry problem with an initial reentry velocity near 26,000 fps is essentially the final phase which is entered into on the initial Apollo entry at velocities less than 27,000 fps.[G-1], and where the closed loop guidance law is a function of perturbations about a pre-stored nominal reference trajectory with velocity as the independent variable.

The method of adjoints[M-4, L-2] was utilized to generate such a final phase reference trajectory for the Apollo reentry, with the adjoint variables becoming the perturbation feedback gains in the control law. The control is expressible as a commanded L/D which specifies the commanded roll angle. The out of plane component of lift is essentially nulled by a lateral logic scheme whereby the lift vector is switched from one side of the trajectory plane to the other. This reference trajectory was also chosen to match a desired down-range so that by following it the terminal error in range could be nulled, while relying on the lateral logic to zero the error in track.

The selection of the Apollo nominal reference reentry trajectory was greatly influenced by operational considerations. The employment of a reference trajectory was made to render the overall reentry guidance and navigation system less sensitive to large input errors, and thus less susceptible to blunderous decisions on the part of the steering logic. Of course, the obvious disadvantage of using a pre-computed reference trajectory is in the reduction of its capability and flexibility to adapt to markedly off design conditions. Here maximum flexibility has been sacrificed in favor of computational simplicity and efficiency.

2.4 The STS Orbiter Reentry Concept

The deorbit burn maneuver for the STS orbiter is designed nominally to occur from a 55 degree inclination, 270 nautical mile circular orbit. The vehicle then enters the atmosphere at a proposed 60 degree angle of attack. The reasons for this high angle of attack are as follows. One major benefit is a reduction in the thermal protection system required. High angle of attack reentry in the region of peak heating and maximum acceleration greatly reduces the heat protection requirements for the lower surfaces of the body. Being in the proximity of the maximum lift coefficient yields low reentry load factors and low peak heating rates. Such a high drag configuration results in a shorter reentry time duration leading to a lower total heat accumulation. Vehicle stability considerations also favor such a high angle of attack blunt body reentry from orbit.

The equations of motion are as indicated in Eqs. (2.2.8) - (2.2.11), with all of the terms retained. The actual control is the trajectory plane or vertical lift or L/D. But these are functions of both the angle of attack which determines the maximum capable amount of lift, and the roll angle which determines the trajectory plane component of this available lift. The redundancy or overlap here involves a trade-off which is influenced by vehicle configuration stability considerations. One method of eliminating this apparent redundancy is to consider the in-plane vertical lift as the principal control variable. However an additional parameter is required in order to evaluate the total acceleration. This is the roll angle. The angle of attack and roll or bank angle are also natural control quantities for such a conventional airplane type vehicle as the STS orbiter.

The investigation for the STS configuration is similar to the final phase analysis of Apollo, employing a perturbation scheme about a reference nominal trajectory. The difference lies in the fact that not only will first order perturbations be included in the control low, but cross coupled terms and second order terms will be taken into account depending on the values of their influence functions. These influence functions are direct results of the optimization process that yields the optimal nominal reference trajectories. A cost functional or performance index is fashioned to include quadratic penalty terms for heat and load limitations and terminal inaccuracies. The object is to reduce this cost functional with respect to the control variables, angle of attack and roll angle. Stability considerations may be introduced either in the form of penalty functions for activity in unstable regions (soft constraints) or as hard constraints which are not to be violated.

CHAPTER III

A DIFFERENTIAL DYNAMIC PROGRAMMING APPROACH

3.1 Differential Dynamic Programming

For a dynamic system described by the non-linear ordinary differential equations or system state equations of the form

$$\dot{\underline{x}}(t) = \underline{f}\left[\underline{x}(t), \underline{u}(t), t\right] \tag{3.1.1}$$

with

$$\underline{x}(t_0) = \underline{x}_0 \tag{3.1.2}$$

the criterion of optimality is the minimization of the cost functional or performance index

$$J(\underline{x}_0, t_0) = \int_{t_0}^{t_f} L\left[\underline{x}(t), \underline{u}(t), t\right]dt + F\left[\underline{x}(t_f), t_f\right] \tag{3.1.3}$$

where $L\left[\underline{x}(t), \underline{u}(t), t\right]$ can be thought of as the accrued cost along the trajectory from time t_0 to t_f, and $F\left[\underline{x}(t_f), t_f\right]$ as the terminal cost. Both are sealar functions of the indicated variables.

Generalizing to any arbitrary starting point $\underline{x}(t)$, and proceeding optimally to a terminal hypersurface defined by:

$$\psi\left[\underline{x}(t_f), t_f\right] = 0 \tag{3.1.4}$$

in the manner of Bryson and Ho[B-1], the unique optimal value of the performance index is the optimal return function

$$J^o = J^o\left[\underline{x}(t),\ t\right] \tag{3.1.5}$$

or

$$J^o\left[\underline{x}(t),\ t\right] = \min_{\underline{u}(t)} \left\{F\left[\underline{x}(t_f),\ t_f\right] + \int_{t_0}^{t_f} L\left[\underline{x}(\tau),\ \underline{u}(\tau),\ \tau\right] d\tau\right\} \tag{3.1.6}$$

with the boundary condition

$$J^o\left[\underline{x}(t),\ t\right] = F\left[\underline{x}(t),\ t\right] \tag{3.1.7}$$

on the hypersurface $\psi\left[\underline{x}(t),\ t\right] = 0$. The superscript o denotes optimal.

To derive the partial differential equation satisfied by the optimal return function, assume that $J^o\left[\underline{x}(t),\ t\right]$ exists, is continuous, and possesses continuous first and second partial derivatives at the points of interest. If the starting point is $\underline{x}(t)$, employment of a non-optimal control $\underline{u}(t)$, over a small time interval Δt moves the system to another point described by

$$\underline{x}(t) + \underline{f}\left[\underline{x}(t),\ \underline{u}(t),\ t\right]\Delta t \tag{3.1.8}$$

at time $t + \Delta t$. Utilizing optimal control from this point on, the return function is written to first order

$$J^1\left[\underline{x}(t),\ t\right] = J^o\left[\underline{x}(t) + \underline{f}\left[\underline{x}(t),\ \underline{u}(t),\ t\right]\Delta t,\ t + \Delta t\right]$$
$$+ L\left[\underline{x}(t),\ \underline{u}(t),\ t\right]\Delta t \tag{3.1.9}$$

Since a non-optimal control is employed over the time interval Δt, it is true that

$$J^o\left[\underline{x}(t),\ t\right] \le J^1\left[\underline{x}(t),\ t\right] \qquad (3.1.10)$$

If $\underline{u}(t)$ is chosen to minimize the right hand side of Eq. (3.1.10) as given in Eq. (3.1.9), we obtain

$$J^o\left[\underline{x}(t),\ t\right] = \min_{\underline{u}(t)} \left\{ J^o\left[\underline{x}(t) + \underline{f}\left[\underline{x}(t),\ \underline{u}(t),\ t\right]\Delta t,\ t + \Delta t\right] + L\left[\underline{x}(t),\ \underline{u}(t),\ t\right]\Delta t \right\} \qquad (3.1.11)$$

Expansion in a Taylor series about $\underline{x}(t)$ and t leads to

$$J^o\left[\underline{x}(t),\ t\right] = \min_{\underline{u}(t)} \left\{ J^o\left[\underline{x}(t),\ t\right] + \frac{\partial J^o}{\partial \underline{x}}\,\underline{f}\left[\underline{x}(t),\ \underline{u}(t),\ t\right]\Delta t + \frac{\partial J^o}{\partial t}\Delta t + L\left[\underline{x}(t),\ \underline{u}(t),\ t\right]\Delta t \right\} \qquad (3.1.12)$$

Taking the terms which do not depend explicitly on $\underline{u}(t)$ out of the braces and letting Δt approach zero, we obtain

$$-\frac{\partial J^o}{\partial t} = \min_{\underline{u}(t)} \left\{ L\left[\underline{x}(t),\ \underline{u}(t),\ t\right] + \frac{\partial J^o}{\partial \underline{x}}\,\underline{f}\left[\underline{x}(t),\ \underline{u}(t),\ t\right] \right\} \qquad (3.1.13)$$

After defining the Hamiltonian

$$H\left[\underline{x}(t),\ \underline{u}(t),\ \frac{\partial J^o}{\partial \underline{x}},\ t\right] = L\left[\underline{x}(t),\ \underline{u}(t),\ t\right] + \frac{\partial J^o}{\partial \underline{x}}\,\underline{f}\left[\underline{x}(t),\ \underline{u}(t),\ t\right] \qquad (3.1.14)$$

we can write the Hamilton-Jacobi-Bellman partial differential equation as follows:

$$-\frac{\partial J^o}{\partial t} = H^o\left[\underline{x}(t), \frac{\partial J^o}{\partial \underline{x}}, t\right] \qquad (3.1.15)$$

along with the boundary condition of Eq. (3.1.7), where

$$H^o\left[\underline{x}(t), \frac{\partial J^o}{\partial \underline{x}}, t\right] = \min_{\underline{u}(t)} \left\{ H\left[\underline{x}(t), \underline{u}(t), \frac{\partial J^o}{\partial \underline{x}}, t\right]\right\} \qquad (3.1.16)$$

Equation (3.1.16) yields the $\underline{u}(t) = \underline{u}^o(t)$ which minimizes the Hamiltonian

$$H\left[\underline{x}(t), \underline{u}(t), \frac{\partial J^o}{\partial \underline{x}}, t\right]$$

in the global sense, while holding $\underline{x}(t)$, $\frac{\partial J^o}{\partial \underline{x}}$, and t fixed. This is equivalent to the Minimum Principle. In the formulation, the quantities

$$\frac{\partial J^o}{\partial t} \quad \text{and} \quad \frac{\partial J^o}{\partial \underline{x}}$$

are understood to be evaluated at $[\underline{x}^o(t), t]$.

The quantity which we are seeking to determine is the optimal control history $\underline{u}^o(t)$, $t \in [t_0, t_f]$. Applying a nominal control $\overline{\underline{u}}(t)$, $t \in [t_0, t_f]$, to the system differential Eq. (3.1.1) subject to the initial conditions of Eq. (3.1.2), and integrating yields a nominal state history $\overline{\underline{x}}(t)$, $t \in [t_0, t_f]$. Equation (3.1.3) is utilized to calculate the nominal cost, $\overline{J}(\underline{x}_0, t_0)$, for this nominal trajectory. The nominal histories include the first initial estimate and all the subsequent trajectories generated and updated by the state equations. Each iterative updating procedure yields a nominal trajectory that approaches closer to the optimum. The final nominal updating should produce the optimal solution to within a prescribed tolerance. If perturbations of the state and control variables are introduced about the nominal state and control histories in the manner described by Jacobson's paper[J-2] according to

$$\underline{x}(t) = \overline{\underline{x}}(t) + \delta\,\underline{x}(t) \tag{3.1.17}$$

$$\underline{u}(t) = \overline{\underline{u}}(t) + \delta\,\underline{u}(t) \tag{3.1.18}$$

where $\delta\underline{x}(t)$ and $\delta\underline{u}(t)$ are not necessarily small quantities, the system state equations, the cost functional, and the Hamilton-Jacobi-Bellman equation are exactly represented by:

$$\frac{d}{dt}\left[\overline{\underline{x}}(t) + \delta\underline{x}(t)\right] = \underline{f}\left[\overline{\underline{x}}(t) + \delta\underline{x}(t),\ \overline{\underline{u}}(t) + \delta\underline{u}(t),\ t\right] \tag{3.1.19}$$

with

$$\overline{\underline{x}}(t_0) + \delta\underline{x}(t_0) = \underline{x}_0 \tag{3.1.20}$$

$$J(\underline{x}_0, t_0) = \int_{t_0}^{t_f} L\left[\overline{\underline{x}}(t) + \delta\underline{x}(t),\ \overline{\underline{u}}(t) + \delta\underline{u}(t),\ t\right]dt + F\left[\overline{\underline{x}}(t_f) + \delta\underline{x}(t_f),\ t_f\right] \tag{3.1.21}$$

and

$$-\frac{\partial J^o}{\partial t}(\overline{\underline{x}}(t) + \delta\underline{x}(t),\ t) = \min_{\delta\underline{u}(t)}\left\{L\left[\overline{\underline{x}}(t)+\delta\underline{x}(t),\ \overline{\underline{u}}(t)+\delta\underline{u}(t),\ t\right] + \frac{\partial J^o}{\partial \underline{x}}(\overline{\underline{x}}(t)+\delta\underline{x}(t), t)\,\underline{f}\left[\overline{\underline{x}}(t)+\delta\underline{x}(t),\ \overline{\underline{u}}(t)+\delta\underline{u}(t),\ t\right]\right\} \tag{3.1.22}$$

The nominal trajectory is now the reference trajectory about which the perturbations are taken.

If the optimal cost functional is sufficiently smooth to permit a power series expansion in $\delta \underline{x}(t)$ about $x(t)$, and the difference between the optimal cost functional and the nominal cost functional is expressed as

$$a^{o}\left[\underline{\overline{x}}(t), t\right] = J^{o}\left[\underline{\overline{x}}(t), t\right] - \overline{J}\left[\underline{\overline{x}}(t), t\right] \tag{3.1.23}$$

where the optimal cost functional $J^{o}\left[\underline{\overline{x}}(t), t\right]$ results from employing the optimal control

$$\underline{u}^{o}(\tau) = \underline{\overline{u}}(\tau) + \delta \underline{u}^{o}(\tau), \quad \tau \epsilon \left[t, t_f\right] \tag{3.1.24}$$

and the nominal cost functional $\overline{J}\left[\underline{x}(t), t\right]$ obtains by using the nominal control

$$\underline{\overline{u}}(\tau), \quad \tau \epsilon \left[t, t_f\right] \tag{3.1.25}$$

the Hamilton-Jacobi-Bellman equation as expressed by Eq. (3.1.22) may be rewritten in the following form:

$$- \frac{\partial \overline{J}}{\partial t} - \frac{\partial}{\partial t} a\left[\underline{\overline{x}}(t), t\right] - \frac{\partial}{\partial t}(J_{\underline{x}})\, \delta \underline{x}(t) - \frac{1}{2} \delta \underline{x}(t)^{T} \frac{\partial}{\partial t}(J_{\underline{x}\,\underline{x}})\, \delta \underline{x}(t)$$
$$\cong \min_{\delta \underline{u}(t)} \left\{ L\left[\underline{\overline{x}}(t) + \delta \underline{x}(t), \underline{\overline{u}}(t) + \delta \underline{u}(t), t\right] + (J_{\underline{x}} + J_{\underline{x}\,\underline{x}}\, \delta \underline{x}(t)) \right.$$
$$\left. \underline{f}\left[\underline{\overline{x}}(t) + \delta \underline{x}(t), \underline{\overline{u}}(t) + \delta \underline{u}(t), t\right] \right\} \tag{3.1.26}$$

where some partial differentiations are noted by subscripts, and $\frac{\partial \overline{J}}{\partial t}$, $J_{\underline{x}}$, and $J_{\underline{x}\,\underline{x}}$ are all evaluated at $\left[\underline{\overline{x}}(t), t\right]$. A more detailed derivation is presented in Appendix A.

Equation (3.1.26) is a good approximation only for small values of $\delta \underline{x}(t)$ because of the truncation of the higher order terms in $\delta \underline{x}(t)$. Since $\delta \underline{x}(t_0) = 0$, the $\delta \underline{x}(t)$ in the interval $[t_0, t_f]$ is due solely to the $\delta \underline{u}(t)$ acting through the perturbed state Eq. (3.1.19). The superscript o has also been eliminated, as the modelling of the cost functional locally by an approximate second-order expansion requires a sufficiently small $\delta \underline{x}(t)$. $\delta \underline{u}(t)$ is selected in such a way as to insure only appropriately small values for $\delta \underline{x}(t)$ to allow for optimality to a specified tolerance. Therefore, should any $\delta \underline{x}(t)$ be allowed, the superscript notation denoting true optimality starting from the state given by Eq. (3.1.17), $\underline{x}(t) + \delta \underline{x}(t)$, would not be completely justifiable. The form of the Hamilton-Jacobi-Bellman Eq. (3.1.26) can now be utilized to determine optimal control histories $\underline{u}^o(t), t \epsilon [t_0, t_f]$, by successively improving the current nominal control histories $\underline{u}(t)$, t, $t \epsilon [t_0, t_f]$.

3.2 A New Second-Order Algorithm For Determining Optimal Control For Unconstrained Problems[J-2]

The new second-order differential dynamic programming approach by Jacobson[J-2] is another version of the backward sweep method. One important point is worthy of emphasis. The restriction of maintaining a globally positive definite inverse second partial derivative matrix of the Hamiltonian with respect to the control is lessened by first minimizing $H\left[\underline{x}(t), \underline{u}(t), J_{\underline{x}}, t\right]$ in the control space to yield an improved minimizing control $\underline{u}^*(t)$. All quantities are then evaluated at $\underline{u}^*(t)$. Clearly, requiring $H_{\underline{u}\,\underline{u}}$ to be positive definite for this improved control is much less confining than demanding global convexity of the Hamiltonian throughout the control space. The difficulties due to nonconvex nominal solutions that occur with the backward sweep method are thus averted.

A well known condition of optimality is that

$$H_{\underline{u}} = \left[\underline{x}(t), \underline{u}^*(t), J_{\underline{x}}, t\right] = 0$$

Perturbations in the state and control are then introduced about $\underline{x}(t)$ and $\underline{u}^*(t)$ in order to obtain a linear feedback relationship between $\delta\underline{x}(t)$ and $\delta\underline{u}(t)^\dagger$, while maintaining the necessary condition of optimality.

Returning to the Hamiltonian-Jacobi-Bellman Eq. (3.1.26), we note that it is valid locally in the state space due to small $\delta\underline{x}(t)$, but globally valid in the control space. Setting $\delta\underline{x}(t)$ to zero for the moment, we may realize the required improved minimizing control by minimizing $H\left[\underline{x}(t), \underline{u}(t), J_{\underline{x}}, t\right]$ in the control space to obtain

$$\underline{u}^*(t) = \bar{\underline{u}}(t) + \delta\underline{u}^*(t) \qquad (3.2.1)$$

† $\delta\underline{u}(t)$ now referenced about $\underline{u}^*(t)$, and not about $\bar{\underline{u}}(t)$ as in Eq. (3.1.18).

Upon reintroducing $\delta \underline{x}(t)$ in Eq. (3.1.26), minimality of the right hand side of the Hamilton-Jacobi-Bellman equation is maintained by choosing a suitable $\delta \underline{u}(t)$, but referenced about $\underline{u}^*(t)$ rather than about $\overline{\underline{u}}(t)$. The size of $\delta \underline{u}(t)$ must be limited so that it does not produce large values for $\delta \underline{x}(t)$. This is to insure the validity of the second-order expansion of the performance idex as mentioned previously. The Hamilton-Jacobi-Bellman equation may now be expressed as:

$$- \frac{\partial J}{\partial t} - \frac{\partial}{\partial t}\left\{ a\left[\overline{\underline{x}}(t), t\right]\right\} - \frac{\partial}{\partial t}(J_{\underline{x}})\, \delta \underline{x}(t)$$

$$- \frac{1}{2}\, \delta \underline{x}(t)^T \frac{\partial}{\partial t}(J_{\underline{x}\underline{x}})\, \delta \underline{x}(t) \cong \min_{\delta \underline{u}(t)} \left\{ H\left[\overline{\underline{x}}(t) + \delta \underline{x}(t), \underline{u}^*(t) \right.\right.$$

$$\left.\left. + \delta \underline{u}(t), J_{\underline{x}} + J_{\underline{x}\underline{x}}\, \delta \underline{x}(t), t\right]\right\}$$

(3.2.2)

where the Hamiltonian is as defined in Eq. (3.1.14).

The following necessary optimality condition

$$H_{\underline{u}}\left[\overline{\underline{x}}(t), \underline{u}^*(t), J_{\underline{x}}, t\right] = 0, \qquad (3.2.3)$$

is true, since $\underline{u}^*(t)$ was chosen to minimize H. Expanding the right hand side of Eq. (3.2.2) about $\underline{x}(t)$ and $\underline{u}^*(t)$ according to Eq. (3.1.17),

$$\underline{x}(t) = \overline{\underline{x}}(t) + \delta \underline{x}(t) \qquad (3.2.4)$$

and

$$\underline{u}(t) = \underline{u}^*(t) + \delta \underline{u}(t) \qquad (3.2.5)$$

and differentiating the result with respect to $\delta \underline{u}(t)$ and employing Eq. (3.2.3), we then assume that $\delta \underline{x}(t)$ is sufficiently small to permit equating the coefficient of the first order term in $\delta \underline{x}(t)$ to zero (second-order terms in $\delta \underline{x}(t)$ for the Hamiltonian if $\delta \underline{x}(t)$ and $\delta \underline{u}(t)$ are of the same order). This yields the optimal linear feedback controller

$$\delta \underline{u}(t) = \beta\, \delta \underline{x}(t) \tag{3.2.6}$$

which maintains the necessary condition of optimality for small $\delta \underline{x}(t)$:

$$H_{\underline{u}}\left[\overline{\underline{x}}(t) + \delta \underline{x}(t),\ u^*(t) + \delta \underline{u}(t),\ J_{\underline{x}} + J_{\underline{x}\underline{x}}\, \delta \underline{x}(t),\ t\right] = (0,\ 0) \tag{3.2.7}$$

The term in Eq. (3.2.6) which is chosen to minimize the expanded version of the right hand side of Eq. (3.2.2) is found to be

$$\beta = -\left[H_{\underline{u}\underline{u}}\right]^{-1} (H_{\underline{u}\underline{x}} + \underline{f}_{\underline{u}}^{T} J_{\underline{x}\underline{x}}) \tag{3.2.8}$$

Note that all quantities are to be evaluated at $[\overline{\underline{x}}(t),\ \underline{u}^*(t),\ t]$ unless otherwise specified. Upon further substitution and rearrangement of the expanded version of the right hand side of the Hamilton-Jacobi-Bellman Eq. (3.2.2), and then equating similar powers of $\delta \underline{x}(t)$ with the right hand side of Eq. (3.2.2), we find that, for sufficiently small values of $\delta \underline{x}(t)$,

$$-\frac{\partial \overline{J}}{\partial t} - \frac{\partial a}{\partial t} = H \tag{3.2.9}$$

$$-\frac{\partial}{\partial t}(J_{\underline{x}}) = H_{\underline{x}} + J_{\underline{x}\underline{x}}\, \underline{f} \tag{3.2.10}$$

$$-\frac{\partial}{\partial t}(J_{\underline{x}\underline{x}}) = H_{\underline{x}\underline{x}} + \underline{f}_{\underline{x}}^{T} J_{\underline{x}\underline{x}} + J_{\underline{x}\underline{x}}\, \underline{f}_{\underline{x}} - (H_{\underline{u}\underline{x}} + \underline{f}_{\underline{u}}^{T} J_{\underline{x}\underline{x}})^{T} \left[H_{\underline{u}\underline{u}}\right]^{-1} (H_{\underline{u}\underline{x}} + \underline{f}_{\underline{u}}^{T} J_{\underline{x}\underline{x}}) \tag{3.2.11}$$

where all terms are to be evaluated at $\left[\underline{\bar{x}}(t), \underline{u}^*(t), t\right]$.

Defining full differentials as explained in Appendix B, we write the required reverse differential equations:

$$-\dot{a} = H - H\left[\underline{\bar{x}}(t), \underline{\bar{u}}(t), J_{\underline{x}}, t\right] \qquad (3.2.12)$$

$$-\dot{J}_{\underline{x}} = H_{\underline{x}} + J_{\underline{x}\underline{x}}\left(\underline{f} - \underline{f}\left[\underline{\bar{x}}(t), \underline{\bar{u}}(t), t\right]\right) \qquad (3.2.13)$$

$$-\dot{J}_{\underline{x}\underline{x}} = H_{\underline{x}\underline{x}} + \underline{f}_{\underline{x}}^{T} J_{\underline{x}\underline{x}} + J_{\underline{x}\underline{x}}\underline{f}_{\underline{x}} - (H_{\underline{u}\underline{x}} + \underline{f}_{\underline{u}}^{T} J_{\underline{x}\underline{x}})^{T}\left[H_{\underline{u}\underline{u}}\right]^{-1} (H_{\underline{u}\underline{x}} + \underline{f}_{\underline{u}}^{T} J_{\underline{x}\underline{x}}) \qquad (3.2.14)$$

where all quantities are to be evaluated at $\left[\underline{\bar{x}}(t), \underline{u}^*(t), t\right]$ unless otherwise specified. The dot notation implies full differentiation of the indicated quantities with respect to time. Further details are to be found in Appendix B. Equations (3.2.12) - (3.2.14) are the reverse differential equations to be integrated backwards from the terminal time t_f, subject to the following terminal boundary conditions:

$$a(t_f) = 0 \qquad (3.2.15)$$

$$J_{\underline{x}}(t_f) = F_{\underline{x}}\left[\underline{\bar{x}}(t_f), t_f\right] \qquad (3.2.16)$$

$$J_{\underline{x}\underline{x}}(t_f) = F_{\underline{x}\underline{x}}\left[\underline{\bar{x}}(t_f), t_f\right] \qquad (3.2.17)$$

These boundary conditions are derived in part from the fact that

$$J\left[\bar{\underline{x}}(t_f),\ t_f\right] = F\left[\bar{\underline{x}}(t_f),\ t_f\right] \tag{3.2.18}$$

Thus the new control to be applied to the system state equations is expressible in either the form

$$\underline{u}(t) = \underline{u}*(t) + \beta\,\delta\underline{x}(t) \tag{3.2.19}$$

or

$$\underline{u}(t) = \bar{\underline{u}}(t) + \delta\underline{u}^*(t) + \beta\,\delta\underline{x}(t) \tag{3.2.20}$$

or simply

$$\underline{u}(t) = \underline{u}^*(t) + \delta\underline{u}(t) \tag{3.2.21}$$

over the time interval $\left[t_0,\ t_f\right]$.

3.3 A New First-Order Algorithm For Unconstrained Problems

A special case of the second-order method[J-2] expands the performance index or cost functional only to first order:

$$J\left[\underline{\bar{x}}(t) + \delta\underline{x}(t),\ t\right] = \bar{J}\left[\underline{\bar{x}}(t),\ t\right] + a + J_{\underline{x}}\,\delta\underline{x}(t) \tag{3.3.1}$$

Thus, the following reverse differential equations are obtained:

$$-\dot{a} = H - H\left[\underline{\bar{x}}(t),\ \underline{\bar{u}}(t),\ J_{\underline{x}},\ t\right] \tag{3.3.2}$$

$$-\dot{J}_{\underline{x}} = H_{\underline{x}} \tag{3.3.3}$$

where all quantities are again evaluated at $\left[\underline{\bar{x}}(t),\ \underline{u}^*(t),\ t\right]$ unless otherwise specified.

Equations (3.3.2) and (3.3.3) are subject to the terminal boundary conditions

$$a(t_f) = 0 \tag{3.3.4}$$

$$J_{\underline{x}}(t_f) = F_{\underline{x}}\left[\underline{\bar{x}}(t_f),\ t_f\right] \tag{3.3.5}$$

The new control is then found to be

$$\underline{u}(t) = \underline{u}^*(t) \tag{3.3.6}$$

3.4 A New Step Size Adjustment Method[J-2]

The $\delta \underline{x}(t)$ generated on the forward state integrations by the new improved minimizing control variable must be small enough for the second-order expansions to be valid. To insure this, merely scaling $\delta \underline{u}^*(t)$ down by a multiplicative factor ϵ, where $0 < \epsilon \leq 1$, is not permissible, since the new improved minimizing control $\underline{u}^*(t)$ as expressed in Eq. (3.2.1) is already embedded in the reverse differential Eq. (3.2.12) - (3.2.14). Furthermore, the fact that global convexity in the control space is not required of H precludes such a linear interpolation between $\underline{u}(t)$ and $\underline{u}^*(t) = \overline{\underline{u}}(t) + \delta \underline{u}^*(t)$.

If we substitute the new control of Eq. (3.2.21) and Eq. (3.1.17) into the dynamic system differential Eq. (3.1.1), we obtain

$$\frac{d}{dt}\left[\overline{\underline{x}}(t) + \delta \underline{x}(t)\right] = \underline{f}\left[\overline{\underline{x}}(t) + \delta \underline{x}(t),\ \underline{u}^*(t) + \delta \underline{u}(t),\ t\right] \tag{3.4.1}$$

where

$$\overline{\underline{x}}(t_0) + \delta \underline{x}(t_0) = \underline{x}_0 \tag{3.4.2}$$

As before, the $\delta \underline{x}(t)$ generated by Eq. (3.4.1) is due only to $\delta \underline{u}^*(t)$ so that $\delta \underline{x}(t_0) = 0$. An inappropriately large $\delta \underline{x}(t)$ indicates that $\underline{u}^*(t)$ is not the optimal control which we are seeking, to a certain tolerance.

One method of constraining the size of $\delta \underline{x}(t)$ is to restrict the time interval over which Eq. (3.4.1) may be integrated. If the nominal state history $\overline{\underline{x}}(t)$ is followed from t_0 to t_1, where $t_1 \epsilon \left[t_0, t_f\right]$, then $\delta \underline{x}(t) = 0$ for $t \epsilon \left[t_0, t_1\right]$. Integrating Eq. (3.4.1) over a short enough time interval $\left[t_1, t_f\right]$ generates a sufficiently small $\delta \underline{x}(t)$ to maintain the validity of the second-order expansions. This is true for any value of $\delta \underline{u}^*(t)$. We consider $\delta \underline{x}(t)$ to be sufficiently small enough provided that the predicted improvement in cost over the time interval $\left[t_1, t_f\right]$,

$$\left| a\left[\bar{\underline{x}}(t_1),\ t_1\right]\right| = \left| \int_{t_f}^{t_1} \left\{ H - H\left[\bar{\underline{x}}(t),\ \underline{u}(t),\ J_{\underline{x}},\ t\right]\right\} dt \right| \tag{3.4.3}$$

employing the control $\underline{u}(t) = \underline{u}^*(t) + \delta\underline{u}(t) = \underline{u}^*(t) + \beta(t)\,\delta\underline{x}(t)$, $t \in \left[t_1,\ t_f\right]$, is sufficiently near the actual improvement in cost over the same time interval:

$$\Delta J = \bar{J}\left[\bar{\underline{x}}(t_1),\ t_1\right] - J\left[\bar{\underline{x}}(t_1),\ t_1\right] \tag{3.4.4}$$

Sufficiently near entails utilizing a "nearness criteria" which is satisfied when

$$\frac{\Delta J}{a\left[\bar{\underline{x}}(t_1),\ t_1\right]} > C \tag{3.4.5}$$

where $0 \leq C \leq 1$. It is the acceptance or rejection judgement here which determines an appropriate t_1 and plays a major role in the general efficiency and effectiveness of the overall optimization procedure.

To subdivide the time interval $\left[t_0,\ t_f\right]$, we start with $t_1 = t_0$, and if a reasonable reduction in cost ΔJ is obtained, we proceed with the main algorithm. If a reasonable ΔJ has not been found, t_1 is updated in the following manner:

$$t_1 = t_{0,r+1} = .5\,(t_f + t_{0,r}) \tag{3.4.6}$$

where $t_{0,0} = -t_f$ to allow $t_{0,1}$ to be equal to $t_0 = 0$.

t_1 must never be allowed to fall into an interval $\left[t_2,\ t_f\right]$, $t_2 \in \left[t_0,\ t_f\right]$, where the nominal trajectory is the optimal trajectory over $\left[t_2,\ t_f\right]$, but not optimal over $\left[t_0,\ t_f\right]$. This would yield a $\delta\underline{x}(t) = 0$ for $t \in \left[t_1,\ t_f\right]$, since $\underline{u}^*(t) = \bar{\underline{u}}(t)$ for $t \in \left[t_2,\ t_f\right]$. No reduction cost would ensue even though the trajectory is non-optimal over the whole time interval $[t_0, t_f]$. Realizing that $a[\underline{x}(t_f), t_f] = 0$, we may record the time $t = t_{eff}$ at which $a[\underline{x}(t), t]$ differs from zero or in practice becomes greater than a small positive quantity, η, during the reverse integration. Thus

$$a\left[\underline{x}(t), t\right] = 0 \quad \text{for} \quad t \in \left[t_{eff}, t_f\right].$$

Equation (3.4.6) is then rewritten for this particular case as:

$$t_1 = t_{0,r+1} = .5\,(t_{eff} + t_{0,r}) \tag{3.4.7}$$

where $t_{0,0} = -t_{eff}$.

Now the interval $[t_0, t_{eff}]$ rather than $[t_0, t_f]$ is subdivided to find t_1. As the nominal trajectories become more nearly optimal, the interval where $a[\underline{x}(t), t] = 0$ approaches the total time interval $[t_0, t_f]$ so that in effect t_{eff} approaches t_0. On the optimal trajectory we then have

$$\left|a\left[\underline{x}(t), t\right]\right| < \eta, \quad t \in \left[t_0, t_f\right] \tag{3.4.8}$$

and $t_{eff} = t_0$.

During the reverse integration process, the variables of integration may become unbounded at some time t_b, $t_b \in [t_0, t_{eff}]$. In this case, the time interval $[t_b, t_{eff}]$ is subdivided to determine t_1 according to

$$t_1 = t_{0,r+1} = .5\left[t_{eff} + t_{0,0}\right] \tag{3.4.9}$$

where $t_{0,0} = 2t_b - t_{eff}$ to insure that $t_{0,1} = t_b$. Such a strategm aids in avoiding conjugate points or instabilities that frequently arise in the backward integration of the reverse matrix Riccati differential Eq. (3.2.14).

Again, as the nominal trajectory approaches the optimal trajectory, t_b approaches t_0. In the end $t_b = t_0$ so that the variables of the reverse integration are bounded over the total time interval of interest $[t_0, t_f]$.

CHAPTER IV

FORMULATION OF THE STS REENTRY PROBLEM

4.1 Equations of Motion and Heating Relationships

For the non-linear STS lifting-body reentry problem the equations of motion are as given by Eqs. (2.2.8) - (2.2.11):

$$\dot{x}_r = \frac{v \cos \gamma}{(1 + h/R_e)} \tag{4.1.1}$$

$$\dot{v} = \frac{-q}{W/S} g C_D + \left(-g + \frac{v^2}{(R_e + h)}\right) \sin \gamma \tag{4.1.2}$$

$$\dot{\gamma} = \frac{q/v}{W/S} g C_L + \left(-g/v + \frac{v}{(R_e + h)}\right) \cos \gamma \tag{4.1.3}$$

$$\dot{h} = v \sin \gamma \tag{4.1.4}$$

where

$$q = \frac{1}{2} \rho v^2 \tag{4.1.5}$$

is the dynamic pressure,

$$D = \frac{1}{2} \rho v^2 S C_D \tag{4.1.6}$$

is the drag force, and

$$L = \frac{1}{2} \rho v^2 S C_L \tag{4.1.7}$$

is the lift force in the vertical trajectory plane. The maximum coefficient of lift and the corresponding maximum lift for a particular angle of attack and Mach number are related to their vertical components, respectively, by:

$$C_L = C_{L_{max}} \cos \phi \tag{4.1.8}$$

$$L = L_{max} \cos \phi \tag{4.1.9}$$

where ϕ is the roll angle about the relative wind axis. Introducing a convenient vector notation, we have

$$\underline{x}(t) = \begin{bmatrix} x_r(t) \\ v(t) \\ \gamma(t) \\ h(t) \end{bmatrix} \tag{4.1.10}$$

for the state vector, and

$$\underline{u}(t) = \begin{bmatrix} \phi(t) \\ \alpha(t) \end{bmatrix} \tag{4.1.11}$$

for the control vector. The state equations are then written as:

$$\dot{\underline{x}}(t) = \begin{bmatrix} \dot{x}_r(t) \\ \dot{v}(t) \\ \dot{\gamma}(t) \\ \dot{h}(t) \end{bmatrix} = \underline{f}\left[\underline{x}(t), \underline{u}(t), t\right] \tag{4.1.12}$$

where $\dot{x}_r$, $\dot{v}$, $\dot{\gamma}$, and $\dot{h}$ are as depicted in Eqs. (4.1.1) - (4.1.4).

The convective heat rate relationship is given by Eq. (2.2.12):

$$\dot{q}_c = \frac{17,600}{\sqrt{R_N}} \left(\frac{\rho}{\rho_0}\right)^{0.5} \left(\frac{v}{26,000}\right)^{3.15} \tag{4.1.13}$$

or

$$\dot{q}_c = .61433466\,(R_N)^{-0.5}\,(\rho/\rho_0)^{0.5}\left(\frac{v}{1000}\right)^{3.15} \tag{4.1.14}$$

for the nose stagnation area in units of BTU/ft^2/sec., where $v = v_\infty$ is the free stream velocity. Radiative heating is neglected for the velocities of interest which are less than 30,000 fps. The total convective heat accumulation is simply the integral of the heat flux, denoted by:

$$q_c = \int_{t_0}^{t_f} \dot{q}_c \, dt \tag{4.1.15}$$

in units of BTU/ft^2.

4.2 The Quadratic Cost Functional

The performance index or cost functional to be minimized is given by Eq. (3.1.3):

$$J(\underline{x}_0, t_0) = \int_{t_0}^{t_f} L\left[\underline{x}(t), \underline{u}(t), t\right] dt + F\left[\underline{x}(t_f), t_f\right] \tag{4.2.1}$$

where the accumulated cost along the trajectory is denoted by the quadratic

$$L\left[\underline{x}(t), \underline{u}(t), t\right] = \frac{1}{2}\begin{bmatrix} \delta\dot{q}_c \\ \delta g's \\ \delta q \\ \sqrt{\widetilde{\delta\dot{q}}_c} \end{bmatrix}^T \begin{bmatrix} W\dot{q}_c & 0 & 0 & 0 \\ 0 & Wg's & 0 & 0 \\ 0 & 0 & Wq & 0 \\ 0 & 0 & 0 & 2\sqrt{W\dot{q}_c} \end{bmatrix} \begin{bmatrix} \delta\dot{q} \\ \delta g's \\ \delta q \\ \sqrt{\widetilde{\delta\dot{q}}_c} \end{bmatrix} \tag{4.2.2}$$

or

$$L = \frac{1}{2}\left\{\delta\dot{q}_c\, W\dot{q}_c\, \delta\dot{q}_c + \delta g's\, Wg's\, \delta g's + \delta q\, Wq\, \delta q\right\} + \sqrt{W\dot{q}_c}\, \widetilde{\delta\dot{q}}_c \tag{4.2.3}$$

and the terminal cost is given as:

$$F\left[\underline{x}(t_f),\ t_f\right] = \frac{1}{2}\begin{bmatrix}\delta x_r \\ \delta v \\ \delta\gamma \\ \delta h\end{bmatrix}^T \begin{bmatrix} W_{x_r} & 0 & 0 & 0 \\ 0 & W_v & 0 & 0 \\ 0 & 0 & W_\gamma & 0 \\ 0 & 0 & 0 & W_h \end{bmatrix} \begin{bmatrix}\delta x_r \\ \delta v \\ \delta\gamma \\ \delta h\end{bmatrix} \tag{4.2.4}$$

or

$$F = \frac{1}{2}\left\{\delta x_r\, Wx_r\, \delta x_r + \delta v\, W_v\, \delta v + \delta\gamma\, W_\gamma\, \delta\gamma + \delta h W_h\, \delta h\right\} \tag{4.2.5}$$

The convective heat rate "penalty" is expressed as

$$\delta\dot{q}_c = \dot{q}_c \qquad (4.2.6)^\dagger$$

---

†The delta notation is adopted to allow for the possibility of maximum threshold values for $\dot{q}_c$ below which virtually no penalty is imposed. The expression

$$\delta\dot{q}_c = \begin{cases} \dot{q}_c - \dot{q}_c\big]_{max} & \text{if } \dot{q}_c > \dot{q}_c\big]_{max} \\ 0 & \text{otherwise} \end{cases}$$

would be utilized were it not for the discontinuities in the second derivatives of $\delta\dot{q}_c$ when used in the reverse differential equations. As Eq. (4.2.6) stands, it is equivalent to employing a threshold value $\dot{q}_c\big]_{max} = 0$. Similar comments apply to δg's and δq in Eqs. (4.2.7) and (4.2.10).

where $\dot{q}_c$ is the convective heat reate given by Eq. (4.1.13). Similarly the acceleration or load factor "penalty" is expressed as

$$\delta g's = g's \tag{4.2.7}$$

where the actual acceleration is given in g's by

$$g's = \frac{1}{32.2\,m}\,(L^2_{max} + D^2)^{0.5} \tag{4.2.8}$$

or

$$g's = \frac{q}{W/S}\,(C_D{}^2 + C_{L_{max}}{}^2)^{0.5} \tag{4.2.9}$$

The dynamic pressure "penalty" is expressible as

$$\delta q = q \tag{4.2.10}$$

where q is the dynamic pressure given by Eq. (4.1.5). The total accumulated convective heat "penalty" is expressed as

$$\delta \dot{q}_c(t) = \dot{q}_c(t) \tag{4.2.11}$$

where q_c is the total convective heat accumulated according to Eq. (4.1.15). The terminal downrange inaccuracy "penalty" is given by

$$\delta x_r = x_r - x_r\big]_{desired} \tag{4.2.12}$$

where x_r is the downrange travelled and $x_r\big]_{desired}$ is the downrange desired. Note that positive δx_r denotes overshoot in range, and negative δx_r undershoot. Similarly

$$\delta v = v - v\Big]_{desired} \tag{4.2.13}$$

$$\delta\gamma = \gamma - \gamma\Big]_{\text{desired}} \tag{4.2.14}$$

$$\delta h = h - h\Big]_{\text{desired}} \tag{4.2.15}$$

denote terminal error "penalties" in velocity, flight path angle, and altitude, respectively. The terminal state values desired are not necessarily single valued, as a deadband or range of terminal values may also be acceptable, and thus not penalized. This is a very useful notion for the particular atmospheric reentry problem at hand, as the subsequent powered cruise flight may commence from a specified range of initial conditions on the states.

The penalties are scaled by the weighting factors $W\dot{q}_c$, Wg's, Wg, Wqc, Wx_r, W_v, W_γ, and W_h. Their values are essentially the inverses squared of the acceptable tolerance or variance in each quantity:

$$W\dot{q}_c = \frac{1}{\sigma_{\dot{q}c}^2} \tag{4.2.16}$$

$$Wg's = \frac{1}{\sigma_{g's}^2} \tag{4.2.17}$$

$$Wq = \frac{1}{\sigma_q^2} \tag{4.2.18}$$

$$W_{qc} = \frac{1}{\sigma_{qc}^2} \tag{4.2.19}$$

$$W_{x_r} = \frac{1}{\sigma_{x_r}^2} \tag{4.2.20}$$

$$W_v = \frac{1}{\sigma_v^2} \tag{4.2.21}$$

$$W_\gamma = \frac{1}{\sigma_\gamma^2} \tag{4.2.22}$$

$$W_h = \frac{1}{\sigma_h^2} \tag{4.2.23}$$

However, any of these may be scaled by a multiplicative constant to alter the relative weightings so that large tolerances are weighted less heavily and small tolerances weighted more heavily, or they may even be time dependent or varying with some other variable.

4.3 The Reverse Differential Equations

The reverse differential equations to be integrated backwards from the terminal time t_f are as given by Eqs. (3.2.12) - (3.2.14):

$$-\dot{a} = H - H\left[\bar{\underline{x}}(t), \bar{\underline{u}}(t), J_{\underline{x}}, t\right] \qquad (4.3.1)$$

$$-\dot{J}_{\underline{x}} = H_{\underline{x}} + J_{\underline{x}\underline{x}}(\underline{f} - \underline{f}\left[\bar{\underline{x}}(t), \bar{\underline{u}}(t), t\right]) \qquad (4.3.2)$$

$$-\dot{J}_{\underline{x}\underline{x}} = H_{\underline{x}\underline{x}} + \underline{f}_{\underline{x}}^T J_{\underline{x}\underline{x}} + J_{\underline{x}\underline{x}}\underline{f}_{\underline{x}} - (H_{\underline{u}\underline{x}} + \underline{f}_{\underline{u}}^T J_{\underline{x}\underline{x}})^T \left[H_{\underline{u}\underline{u}}\right]^{-1} (H_{\underline{u}\underline{x}} + \underline{f}_{\underline{u}}^T J_{\underline{x}\underline{x}}) \qquad (4.3.3)$$

where all terms are evaluated at $\left[\bar{\underline{x}}(t), \underline{u}^*(t), t\right]$ unless otherwise indicated. The left hand sides of Eqs. (4.3.1) - (4.3.3) are the total time derivatives of the variables of the backward integrations.

In otder to see how some of the terms on the right hand side appear in this particular problem, we write the Hamiltonian as defined in Eq. (3.1.14):

$$H\left[\underline{x}(t), \underline{u}(t), \frac{\partial J^0}{\partial \underline{x}}, t\right] = L\left[\underline{x}(t), \underline{u}(t), t\right] + \frac{\partial J^0}{\partial \underline{x}} \underline{f}\left[\underline{x}(t), \underline{u}(t), t\right] \qquad (4.3.4)$$

where L is given by Eq. (4.2.3) and $\underline{f}$ by Eq. (4.1.12). $\underline{f}_{\underline{x}}$ is obtained by differentiating Eq. (4.1.12) with respect to the state vector of Eq. (4.1.10), while $\underline{f}_{\underline{u}}$ is the partial of $\underline{f}$ with respect to the control vector of Eq. (4.1.11). Explicit expressions are given in Appendix C.

From Eq. (4.3.4) we assume that $J_{\underline{x}}$ is the classical Lagrange multiplier influence function of the cost functional with respect to variations in the state vector to render

$$H_{\underline{x}} = L_{\underline{x}} + (J_{\underline{x}} \underline{f})_{\underline{x}} \tag{4.3.5}$$

or taking the $J_{\underline{x}}$ out of the differentiation,

$$H_{\underline{x}} = L_{\underline{x}} + J_{\underline{x}} \underline{f}_{\underline{x}} \tag{4.3.6}$$

Similarly

$$\begin{aligned} H_{\underline{x}\underline{x}} &= \left(\left[H_{\underline{x}} \right]^T \right)_{\underline{x}} \\ &= L_{\underline{x}\underline{x}} + \left(\left[J_{\underline{x}} \underline{f}_{\underline{x}} \right]^T \right)_{\underline{x}} \\ &= L_{\underline{x}\underline{x}} + \begin{bmatrix} J_{\underline{x}} \underline{f}_{\underline{x}x_r} \\ J_{\underline{x}} \underline{f}_{\underline{x}v} \\ J_{\underline{x}} \underline{f}_{\underline{x}\gamma} \\ J_{\underline{x}} \underline{f}_{\underline{x}h} \end{bmatrix} \end{aligned} \tag{4.3.7}$$

where the bracketted term is depicted as such in order to avoid third order tensor notation. These quantities are further expounded in Appendix C. In an analogous fashion, we may express

$$H_{\underline{u}} = L_{\underline{u}} + J_{\underline{x}} \underline{f}_{\underline{u}} \tag{4.3.8}$$

$$H_{\underline{u}\underline{u}} = L_{\underline{u}\underline{u}} + \begin{bmatrix} J_{\underline{x}} \underline{f}_{\underline{u}\phi} \\ J_{\underline{x}} \underline{f}_{\underline{u}\alpha} \end{bmatrix} \tag{4.3.9}$$

and

$$H_{\underline{u}\,\underline{x}} = \left(\left[H_{\underline{u}} \right]^T \right)_{\underline{x}}$$

$$= L_{\underline{u}\,\underline{x}} + \begin{bmatrix} J_{\underline{x}} \; \underline{f}_{x\phi} \\ J_{\underline{x}} \; \underline{f}_{x\alpha} \end{bmatrix} \qquad (4.3.10)$$

A detailed derivation of these is given in Appendix C.

The partials of the accrued cost functional L required in Eqs. (4.3.5) - (4.3.10) are given as follows. Recall the expression for L in Eq. (4.2.2) and (4.2.3):

$$L = \frac{1}{2}\left\{ W\dot{q}_c \, (\delta\dot{q}_c)^2 + W_{g's} \, (\delta_{g's})^2 + Wq \, (\delta q)^2 + 2\sqrt{W\dot{q}_c} \, (\widetilde{\delta\dot{q}_c})^2 \right\} \qquad (4.3.11)$$

Taking the desired partial derivatives we find that

$$L_{\underline{x}} = \begin{bmatrix} L_{x_r} \\ L_v \\ L_\gamma \\ L_h \end{bmatrix}^T$$

$$= \begin{bmatrix} 0 \\ \delta\dot{q}_c \, W\dot{q}_c \, (\delta\dot{q}_c)_v + \delta_{g's} \, W_{g's} \, (\delta_{g's})_v + \delta q \, W_q \, (\delta q)_v + \sqrt{W\dot{q}_c} \, (\widetilde{\delta\dot{q}_c})_v \\ 0 \\ \delta\dot{q}_c \, W\dot{q}_c \, (\delta\dot{q}_c)_h + \delta_{g's} \, W_{g's} \, (\delta_{g's})_h + \delta q \, Wq \, (\delta q)_h + \sqrt{W\dot{q}_c} \, (\widetilde{\delta\dot{q}_c})_h \end{bmatrix}^T \qquad (4.3.12)$$

$$L_{\underline{x}\,\underline{x}} = \begin{bmatrix} L_{x_r} \\ L_v \\ L_\gamma \\ L_h \end{bmatrix}_{\underline{x}} \qquad (4.3.13)$$

$$L_{\underline{u}} = \begin{bmatrix} L_\phi \\ L_\alpha \end{bmatrix}^T$$

$$= \begin{bmatrix} \delta_{g's} W_{g's} (\delta_{g's})_\phi \\ \delta_{g's} W_{g's} (\delta_{g's})_\alpha \end{bmatrix}^T \tag{4.3.14}$$

$$L_{\underline{u}\underline{u}} = \begin{bmatrix} L_\phi \\ L_\alpha \end{bmatrix}_{\underline{u}} \tag{4.3.15}$$

$$L_{\underline{u}\underline{x}} = \left(\begin{bmatrix} L_{\underline{u}} \end{bmatrix}^T \right)_{\underline{x}} \tag{4.3.16}$$

These are stated explicitly in Appendix C along with expressions for the partials of the delta terms, $\delta(\ldots)$ and $\widetilde{\delta}(\dot{q}_c)$.

Terminal boundary conditions for the reverse differential Eqs. (4.3.1) - (4.3.3) are given by Eqs. (3.2.15) - (3.2.17):

$$a(t_f) = 0 \tag{4.3.17}$$

$$J_{\underline{x}}(t_f) = F_{\underline{x}} \left[\underline{\bar{x}}(t_f),\ t_f \right] \tag{4.3.18}$$

$$J_{\underline{x}\underline{x}}(t_f) = F_{\underline{x}\underline{x}} \left[\underline{\bar{x}}(t_f),\ t_f \right] \tag{4.3.19}$$

Recalling the form of the terminal cost functional in Eq. (4.2.5):

$$F = \frac{1}{2} \left\{ \delta x_r\, Wx_r\, \delta x_r + \delta v\, W_v\, \delta v + \delta\gamma\, W_\gamma\, \delta\gamma + \delta h\, W_h\, \delta h \right\} \tag{4.3.20}$$

we take the desired partial derivatives required in Eqs. (4.3.18) - (4.3.19):

$$F_{\underline{x}} = \begin{bmatrix} F_{x_r} \\ F_v \\ F_\gamma \\ F_h \end{bmatrix}^T_{t_f}$$

$$= \begin{bmatrix} W_{x_r} & \delta x_r \\ W_v & \delta v \\ W_\gamma & \delta\gamma \\ W_h & \delta h \end{bmatrix}^T_{t_f} \qquad (4.3.21)$$

and

$$F_{\underline{x}\,\underline{x}} = \left\{ \left(\left[F_{\underline{x}} \right]^T \right)_{\underline{x}} \right\}_{t_f} \qquad (4.3.22)$$

Further explication of these boundary conditions may be found in Appendix C.

CHAPTER V

COMPUTATIONAL PROCEDURE

5.1 The Main Algorithm

To solve an optimal control problem such as the non-linear lifting-body atmospheric reentry one by the new differential dynamic programming approach, an initial nominal control history $\bar{u}(t)$, $t \in [t_0, t_f]$, must be available, either from a previous run or from an estimation or initial guess. Integrating the state Eq. (4.1.12) forward from the initial conditions at t_0, and employing the nominal control history $\bar{u}(t)$, a nominal state history is generated, $\bar{x}(t)$, along with other desired quantities and the total cost. Since an initial estimate of the terminal time is also required in this formulation, it may be updated at this point in the algorithm. In this way, it is not incorporated directly into the procedure, but left as a variable parameter to be updated or changed depending on the needs and requirements of the optimization procedure. If the terminal time is increased, the state vector may be extended by maintaining the control vector of the old terminal time constant between the old terminal time and the updated terminal time. For a decrease in terminal time, a mere truncation is all that is required. There is a method[J-3] which incorporates more directly the terminal time updating procedure, but entails increasing the dimension of the problem, and thus increasing the computational burden as well.

The reverse differential Eqs. (4.3.1) - (4.3.3) are then integrated backwards from the boundary conditions of Eqs. (4.3.17) - (4.3.19) to the initial time t_0. It is during this reverse integration process that the Hamiltonian is minimized in the control space to determine $\underline{u}^*(t)$. The optimal and nominal cost difference is monitored during the backward integration, and the time t_{eff} is recorded at which $a\left[\bar{\underline{x}}(t), t\right]$ differs significantly from zero, or in practice when $\left|a\left[\bar{\underline{x}}(t), t\right]\right|$ exceeds a small positive constant, η. The variables of the reverse integrations are also closely monitored, and the time t_b is recorded when the backward integration goes unstable and becomes unbounded, where in general, if t_b exists, $t_0 < t_b < t_{eff} < t_f$.

The new step size adjustment method then determines a new improved nominal trajectory provided it exists and can be found according to the control of Eq. (3.2.19):

$$\underline{u}(t) = \underline{u}^*(t) + \beta\, \delta\underline{x}(t) \qquad (5.1.1)$$

A time t_1, $t_1 \epsilon \left[t_0, t_{eff}\right]$ or $t_1 \epsilon \left[t_b, t_{eff}\right]$ is chosen, where the old nominal is employed for $t \epsilon \left[t_0, t_1\right]$ while an improved optimal is valid over $\left[t_1, t_f\right]$. t_1 is the earliest time at which the actual improvement in the cost functional is sufficiently near the predicted cost improvement. This is expounded in Eqs. (3.4.3) - (3.4.5). t_1 is determined through Eqs. (3.4.6), (3.4.7) and (3.4.9).

If a new improved nominal trajectory cannot be found, the new step size adjustment method stops the computational algorithm. If a new nominal trajectory can be determined, the reverse differential Eqs. (4.3.1) - (4.3.3) are integrated backwards again, and the procedure continues until either an optimal trajectory has been determined to some specified tolerance, or until improvement in the nominal trajectory is no longer attainable. The main algorithm is depicted in the flow diagram of Fig. 5.1.1.

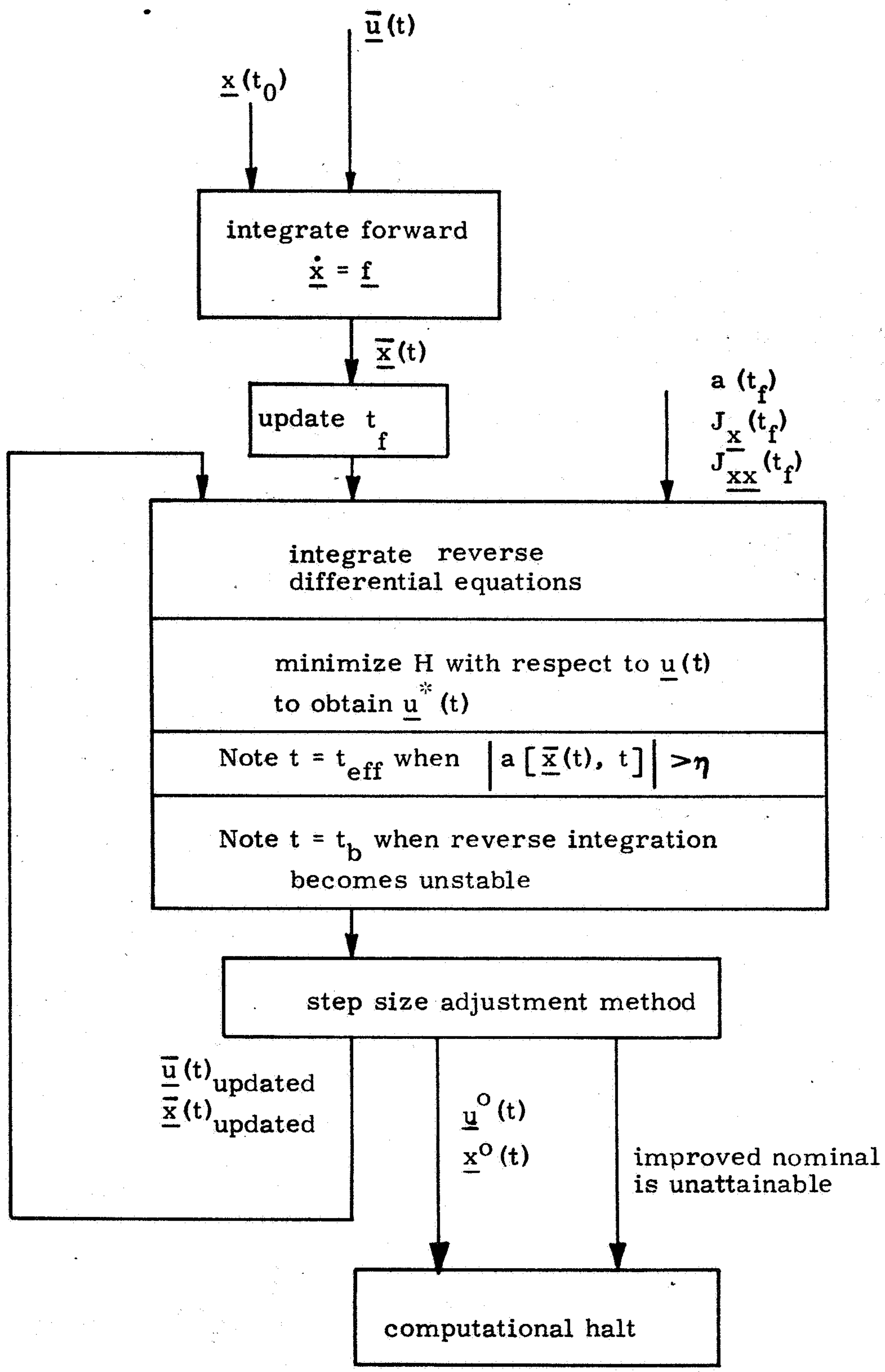


Figure 5.1.1 Main Algorithm

5.2 The New Step Size Adjustment Method

To determine a new improved nominal trajectory, the step size adjustment routine obtains from the main algorithm the time t_{eff} at which the magnitude of $a\left[\underline{\bar{x}}(t), t\right]$ exceeds a small positive constant, η. The optimal trajectory is specified when t_{eff} becomes less than or equal to the initial time t_0. Otherwise the constant c of Eq. (3.4.5) is initially set to 0.5. A t_1 is found according to either Eq. (3.4.7) or (3.4.9) over the time interval $\left[t_0, t_{eff}\right]$ or $\left[t_b, t_{eff}\right]$, respectively, such that an application of the nominal control, $\underline{\bar{u}}(t)$, over $\left[t_0, t_1\right]$ and of the new improved optimal control, $\underline{u}(t) = \underline{u}^*(t) + \delta\underline{u}(t)$, over $\left[t_1, t_f\right]$ leads to a predicted cost improvement sufficiently near the actual cost improvement according to Eq. (3.4.5). When this condition is satisfied, control is transferred back to the main algorithm which proceeds with the next step. Otherwise t_1 is tested to see if it is between t_0 and t_{eff}. If it is, t_1 is updated through Eqs. (3.4.7) or (3.4.9) and the nearness criteria of Eq. (3.4.5) is tested again. If t_1 is not inside the interval $\left[t_0, t_{eff}\right]$, c is set to zero and t_1 is again sought over $\left[t_0, t_{eff}\right]$ or $\left[t_b, t_{eff}\right]$ to satisfy Eq. (3.4.5). If c is already zero, the step size adjustment method halts the computation, as no further improvement of the nominal trajectory can be obtained.

It is to be noted that t_1 is actually tested for inclusion in the interval $\left[t_0, t_{eff} - dt\right]$, where dt is the length of a time step, rather than in the interval $\left[t_0, t_{eff}\right]$. This is to preclude t_1 from ever being updated into the interval $\left[t_{eff}, t_f\right]$ over which interval $\left|a\left[\underline{\bar{x}}(t), t\right]\right| \leq \eta$ and optimality is satisfied, as described in Section 3.4. Figure 5.2.1 exhibits a flow diagram of such a new step size adjustment method.

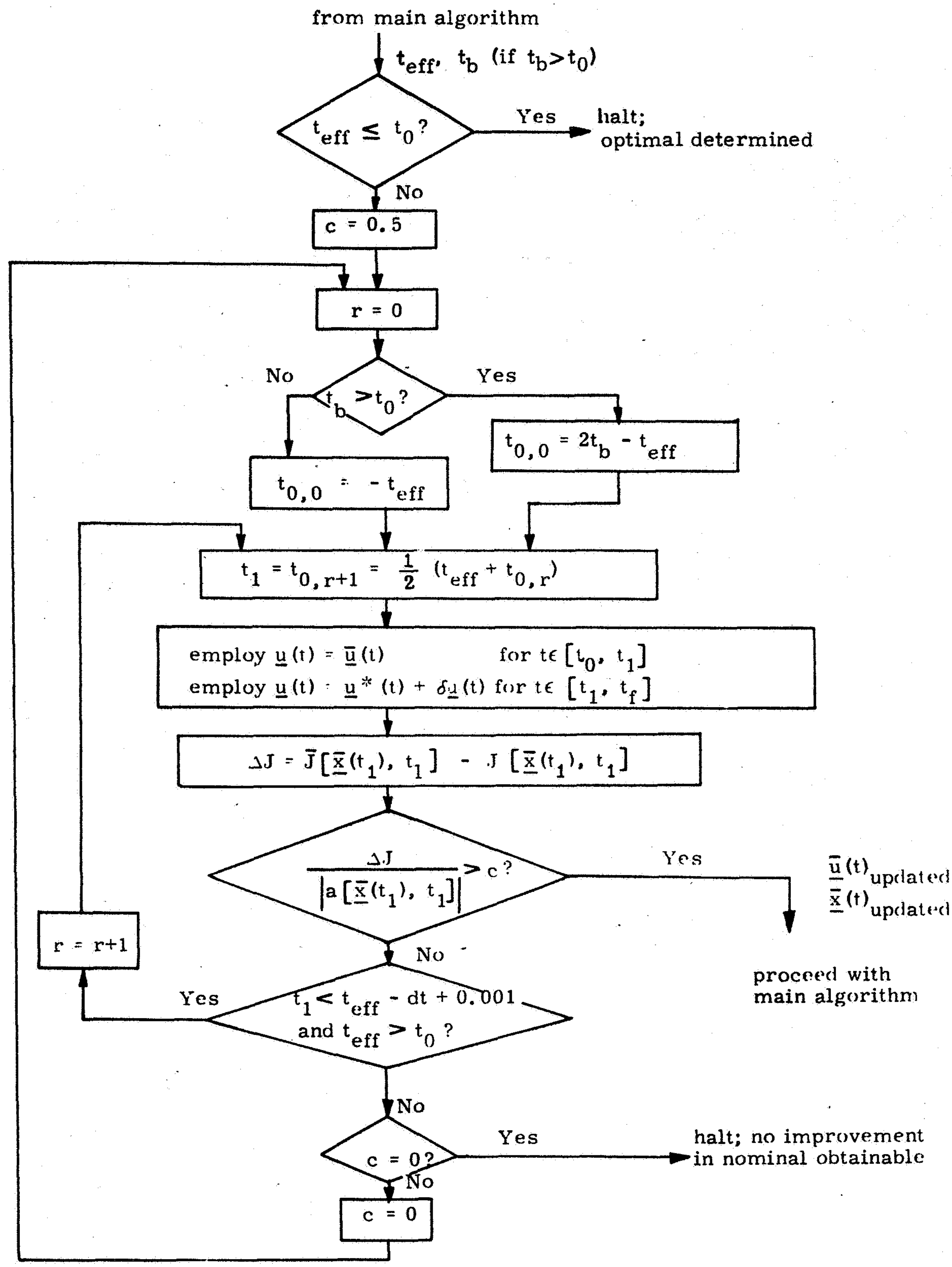


Figure 5.2.1 Step-size Adjustment Method

5.3 A Computational Device for Determining $\underline{u}^*(t)^{J-2}$

The new control to be applied is stated from Eq. (3.2.19):

$$\underline{u}(t) = \underline{u}^*(t) + \beta(t)\,\delta\underline{x}(t) \tag{5.3.1}$$

In non-linear examples, it may be plausible for the second term on the right hand side of Eq. (5.3.1) to be large enough to invalidate the local expansion in the control variables. However, $\delta\underline{x}(t)$ alone may still be sufficiently small such that the following expansion remains justified:

$$J_{\underline{x}}\left[\overline{\underline{x}}(t),\ \delta\underline{x}(t),\ t\right] = J_{\underline{x}}\left[\overline{\underline{x}}(t),\ t\right] + J_{\underline{x}\underline{x}}\left[\overline{\underline{x}}(t),\ t\right]\delta\underline{x}(t) \tag{5.3.2}$$

This being the case, another method may be utilized for purposes of determining the new optimal control $\underline{u}(t)$. Rather than seeking the desired $\delta\underline{u}(t)$ referenced about $\underline{u}^*(t)$, $\underline{u}(t)$ can be gotten directly by minimizing:

$$H\left[\overline{\underline{x}}(t) + \delta\underline{x}(t),\ \underline{u}(t),\ J_{\underline{x}}\left[\overline{\underline{x}}(t),\ t\right] + J_{\underline{x}\underline{x}}\left[\overline{\underline{x}}(t),\ t\right]\delta\underline{x}(t),\ t\right]$$

with respect to $\underline{u}(t)$ either analytically or numerically. Such a technique would also increase the radius of convergence of the algorithm.

In a similar fashion, $\underline{u}^*(t)$ can be determined by setting $\delta\underline{x}(t)$ equal to zero. This would occur during the backward integration of the reverse differential equations.

CHAPTER VI

RESULTS

6.1 On The Formulation of the Algorithm

The reverse differential equations for $J_{\underline{xx}}$ require a nonsingular $H_{\underline{uu}}$ matrix. For the instances when it does become singular, it may be replaced by some predetermined matrix. A truer alternative is utilized in the algorithm of this thesis, especially prevalent in the cases where a constant angle of attack is maintained. During the time steps where $H_{\underline{uu}}$ does become singular, the control vector consisting of roll angle, ϕ, and angle of attack, α, is reduced to a scalar control variable, ϕ. The $H_{\underline{uu}}$ matrix is accordingly replaced by the ϕ-component, $H_{\phi\phi}$, or in other words, by the (1, 1) component of the original $H_{\underline{uu}}$ matrix. Effectively, the influence of α on the Hamiltonian is ignored during this time interval, and only the effect of ϕ is retained, much more representative than replacing the whole matrix by a predetermined value. It is only when $H_{\phi\phi}$ vanishes that a value is substituted for $H_{\phi\phi}$, as opposed to four constant values for the $H_{\underline{uu}}$ matrix. A previous value of $H_{\phi\phi}$ is tried first, provided it is non-zero. Only if this is zero is a predetermined scalar value for $H_{\phi\phi}$ utilized. For these cases, the dimensions of $H_{\underline{ux}}$ and $\underline{f}_{\underline{u}}$ are also changed accordingly.

The range values for ϕ and α are bounded, and although an algorithm does exist for control inequality constrained optimization problems[J-2], a more convenient formulation circumvents the additional computational burdens which it entails. As the data is introduced in the form of a finite data point table, the values assumed outside of the range of interest of the given table are irrelevant and may take on any value. If we hypothesize that "no significant benefits in cost" can be accrued outside of this range of interest, then by testing and sampling in this region only, we are effectively sampling the entire control space, and thus essentially have a simpler unconstrained control formulation. Herein lies an advantage of not having the data as empirical functions of the control variables, and the obvious savings in the dimensionality of the formulation and computation time manifests itself. Another way of viewing this is to realize that only the controls in this region of interest are desirable or acceptable. To ensure that the controls do fall into this specific range of values, large penalties may be imposed on excursions outside of this region. A much simpler approach excludes the undesirable control space from ever being sampled at all, for it would incur such heavy penalties as to be rejected anyway. This "foreknowledge" leads to considerable savings in time during the search of the control space.

The lift and drag coefficient data are functions of both Mach number and angle of attack. The two-dimensional search for these aerodynamic coefficients and their first and second derivatives was first attempted using linear interpolation in each of two directions, and secondly by means of independent four-point binomial local fit along each dimension. When both were found to be inadequate, due to the accuracy requirements for the first and second derivatives with respect to both Mach number and angle of attack, plus a cross derivative, a Newton third order fit was utilized and found to be satisfactory. Since the study was primarily an initial analysis of the c.g. trajectory, functional dependence of C_L and C_D on only angle of attack (and a hypersonic Mach number) is deemed acceptable. This is because the aerodynamic characteristics remain fairly constant until the lower Mach numbers are attained. At these lower Mach numbers, however, the changes in these aerodynamic coefficients are incapable of altering the reentry trajectory to any great extent, thus validating the engineering assumption for such an initial trajectory analysis. For qualitative comparisons with the Apollo reentries, a trim angle of attack of 60^o may be maintained throughout the entire entry, resulting in a constant L/D value of 0.521. In this simplified case, C_L and C_D are simply constant values selected from the data table.

The basic formulation is for a continuous system, and as a digital computer requires a discrete sampled version of the continuous histories of the variables, there arise problems of interpolation of values and synchronization of values held constant over finite time intervals. This is further complicated by the fact that the Runge-Kutta integration routine contains four steps of unequal duration within the overall integration time step. The most feasible interpolation and synchronization procedure was found to be the following. At every full time step, store the sampled state variables from the forward integration of the state equations. These were retrieved at the beginning of each reverse integration step, and held constant throughout the entire step time. Within the reverse integration step, new controls $\underline{u}^*$ were determined utilizing current values of the reverse variables of integration. On the forward integration to determine the new updated nominal control, the values for the reverse variables of integration were retrieved at the start of each full forward integration step and held constant for the remainder of that time step. Within the individual forward integration steps, current values of the states were employed to update the control. In both cases, truer representations of the actual control were rendered, leading to more accurate representations of the sensitive reverse variables of integration. The interpolation and synchronization scheme is contained in the main program of the optimization procedure listed in Appendix F.

6.2 Numerical Results

The differential dynamic programming formulation for optimizing atmospheric reentry trajectories was programmed in the pseudo-language MAC for implementation on the IBM 360/75J. In MAC, all values used in computation are stored as double-precision floating-point numbers occupying double words of eight byte or sixty-four bit length. This is equivalent to an accuracy of 15.9 - 16.8 decimal digits, or in decimal notation, the range of a double-precision floating-point number lies between 10^{-78} and 10^{76}. An introductory description of MAC appears in Appendix E. Listings and explanations of the programs and subroutines employed by the optimization routine appear in Appendix F.

Due to the existence of the nonlinear term in the matrix Riccati reverse differential Eq. (4.3.3), instability during the backward integration is defined as when the magnitude of any component of $J_{\underline{x}\underline{x}}$ exceeds 10^{35}. Any $J_{\underline{x}\underline{x}}$ component exceeding an absolute value of 10^{35} is set to 10^{35} with the appropriate sign, in order to prevent an overflow abort and to allow the proper completion of the integration for that time step. The same magnitude bound on the components of $H_{\underline{u}\underline{u}}$ is also implemented, even though magnitude violations in the absolute values of the components of $H_{\underline{u}\underline{u}}$ do not directly cause an instability in the manner defined above for the $J_{\underline{x}\underline{x}}$ components.

The atmospheric reentry trajectories were integrated forward from the following set of initial conditions:

$$x_r(t_0) = 0 \text{ n.m.} \tag{6.2.1}$$

$$v(t_0) = 26{,}000 \text{ fps} \tag{6.2.2}$$

$$\gamma(t_0) = -1.5 \text{ deg.} \tag{6.2.3}$$

$$h(t_0) = 400{,}000 \text{ ft.} \tag{6.2.4}$$

An exponentially varying atmospheric density model was utilized, with a scale height, h_s, of 28,500 ft. The speeds of sound and its variations with altitude were extracted from the 1962 U.S. Standard Atmosphere[U-1]. Full integration time steps were of 1 second duration, both in the forward and reverse directions. Ordinarily the forward integration of the state equations does not require such a small time step. However the need of the very sensitive reverse integrations to employ state variable values of better accuracy than perhaps interpolated values deemed it more feasible to use the same step size in both directions. Computation time and core memory allocations constituted the major obstacles preventing a desirable further decrease in the size of the reverse integration step. This is with respect to a total time interval of interest on the order of hundreds of seconds. In addition, as control constaints had not yet been included, the angle of attack, α, was constrained to a 60° value for all Mach numbers above 5. Since this was most desirable from an aero-dynamical stability and control point of view, penalizing deviations from a 60° angle of attack above Mach 5 was disregarded in favor of such a simpler fixed constraint approach. In each of the following cases, the total acceleration and dynamic pressure profiles were penalized. The threshold values at which penalization occurs were both zero. Deviations from the desired terminal states were also penalized according to the various weighting schemes given.

The first case is shown in Figs. 6.2.1a - 6.2.1h, and assumes zero weightings excepting the following:

$$\sigma_{g's} = 0.5 \tag{6.2.5}$$

or

$$W_{g's} = 4 \tag{6.2.6}$$

$$\sigma_{x_r} = 1 \tag{6.2.7}$$

or

$$W_{x_r} = 1 \tag{6.2.8}$$

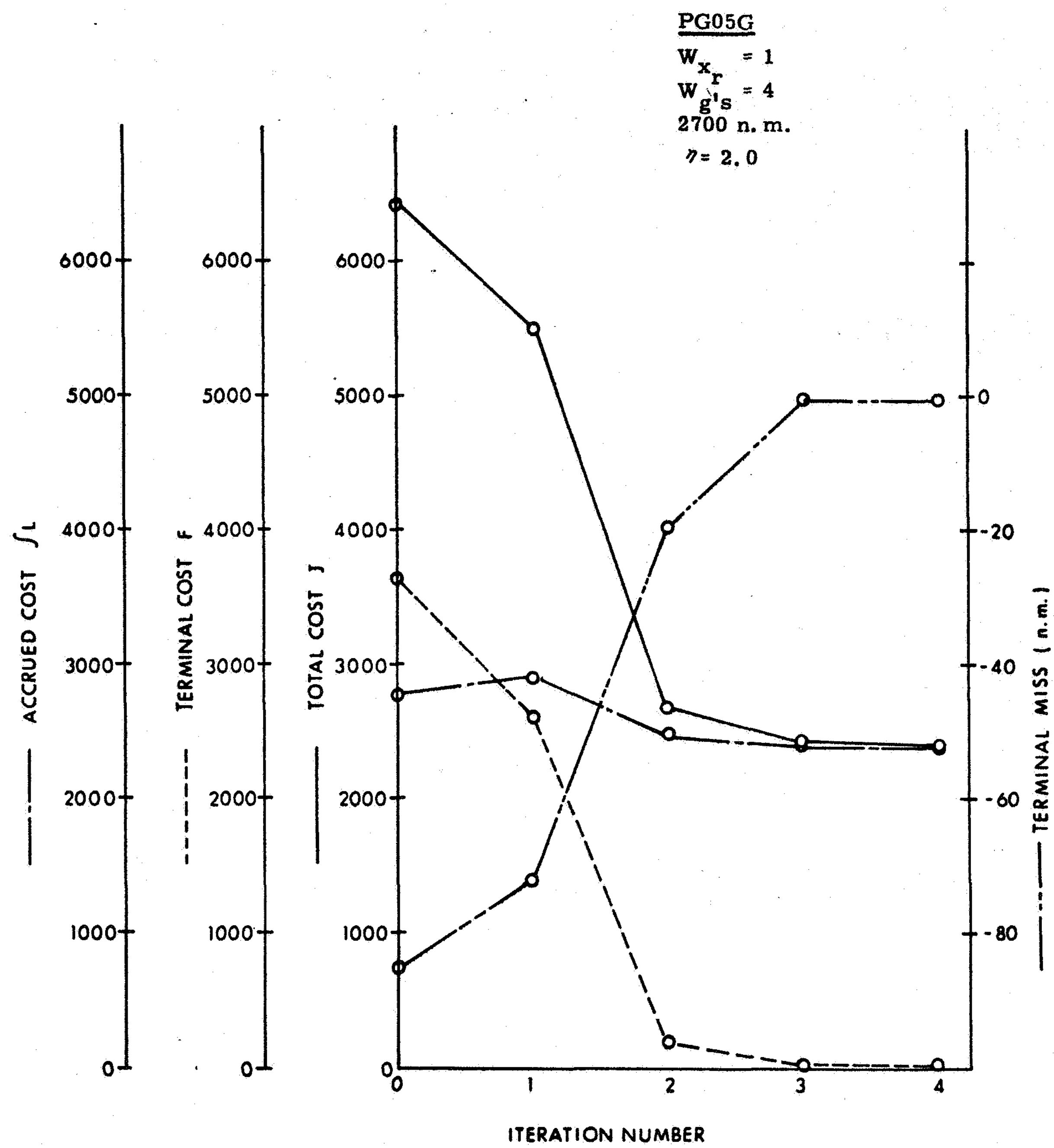

PG05G
W_{x_r} = 1
$W_{g's}$ = 4
2700 n. m.
η = 2. 0
ACCRUED COST $\int L$
TERMINAL COST F
TOTAL COST J
TERMINAL MISS (n. m.)
6000
5000
4000
3000
2000
1000
0
0
-20
-40
-60
-80
0
1
2
3
4
ITERATION NUMBER

Figure 6. 2. 1 a

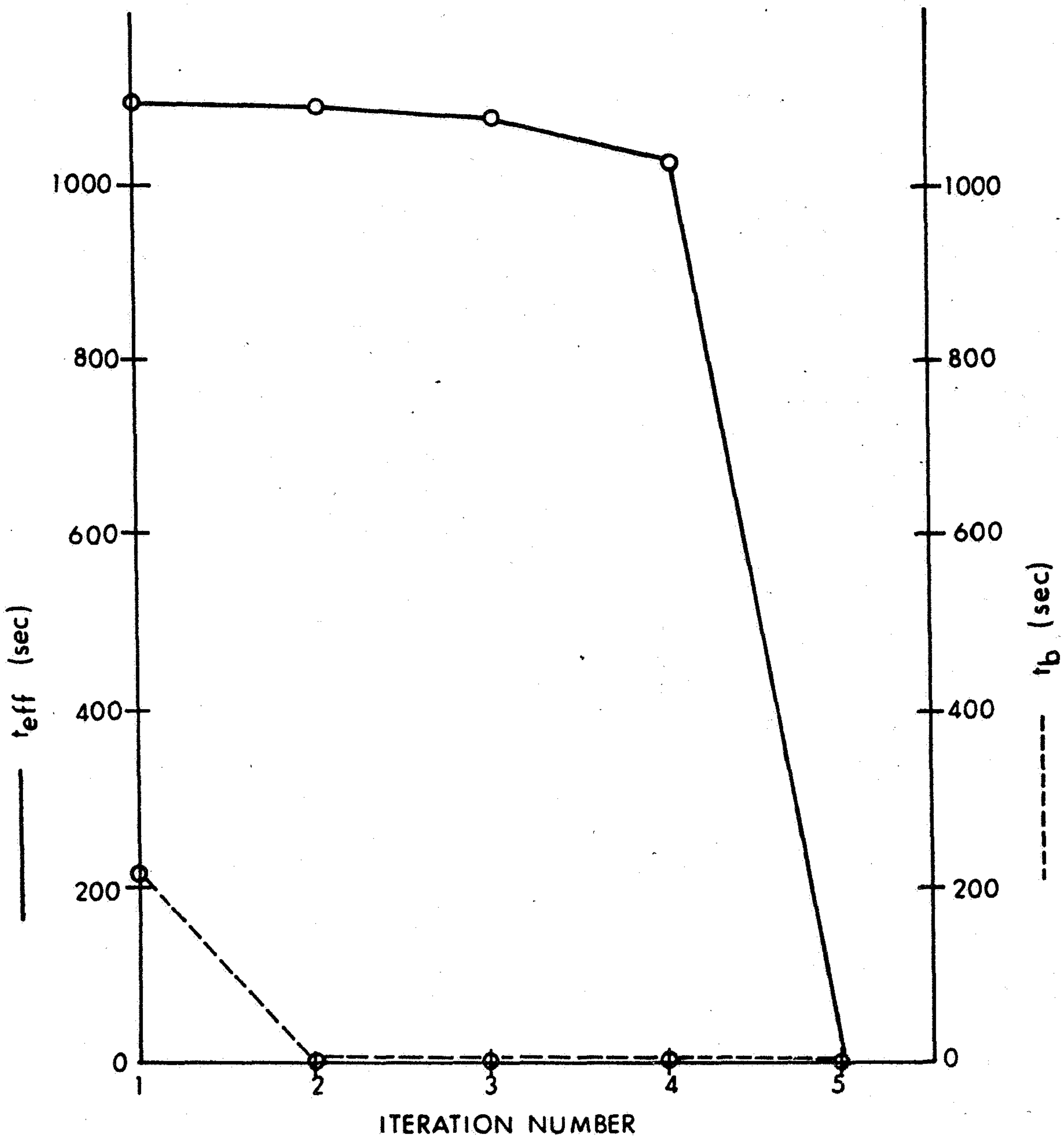
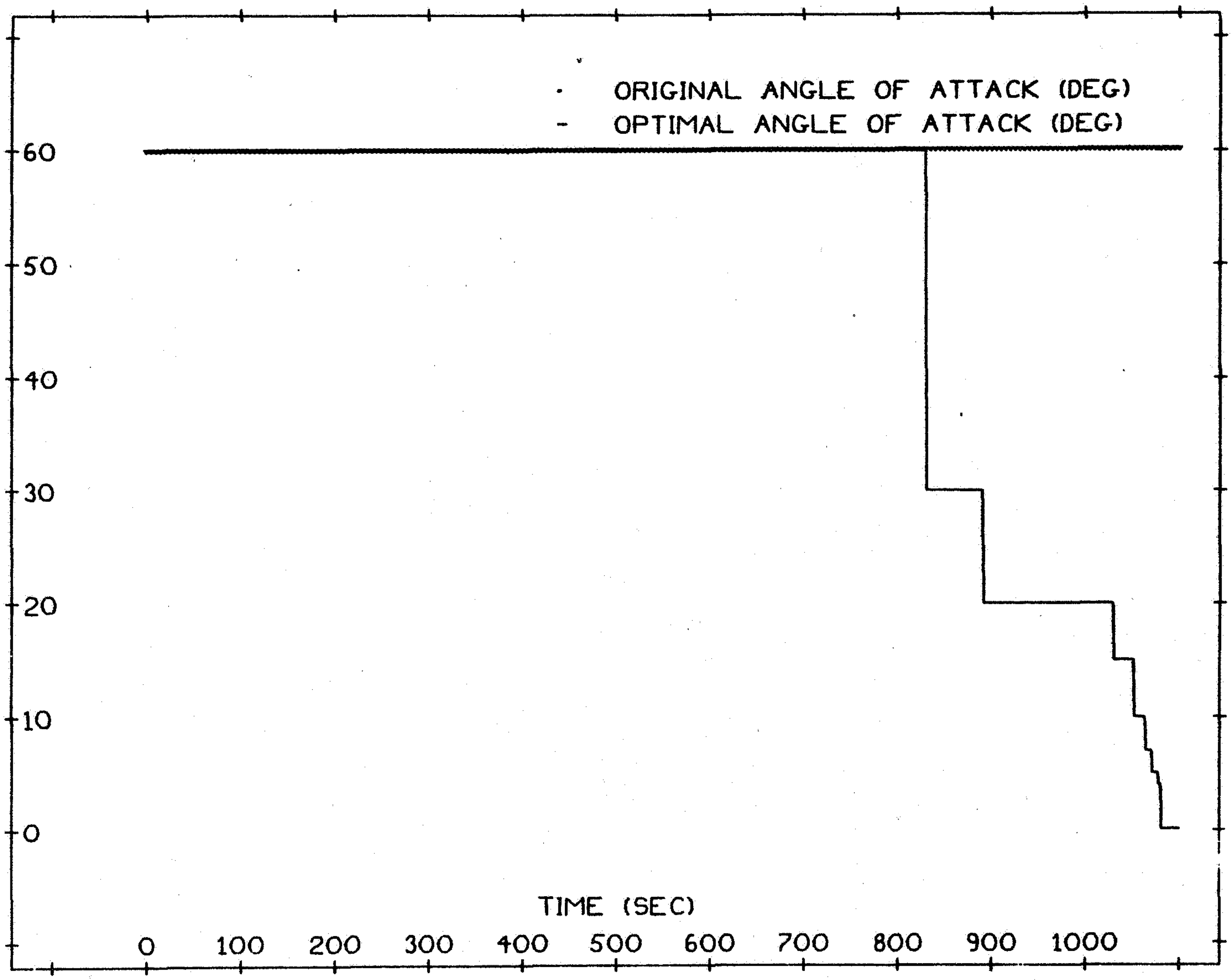
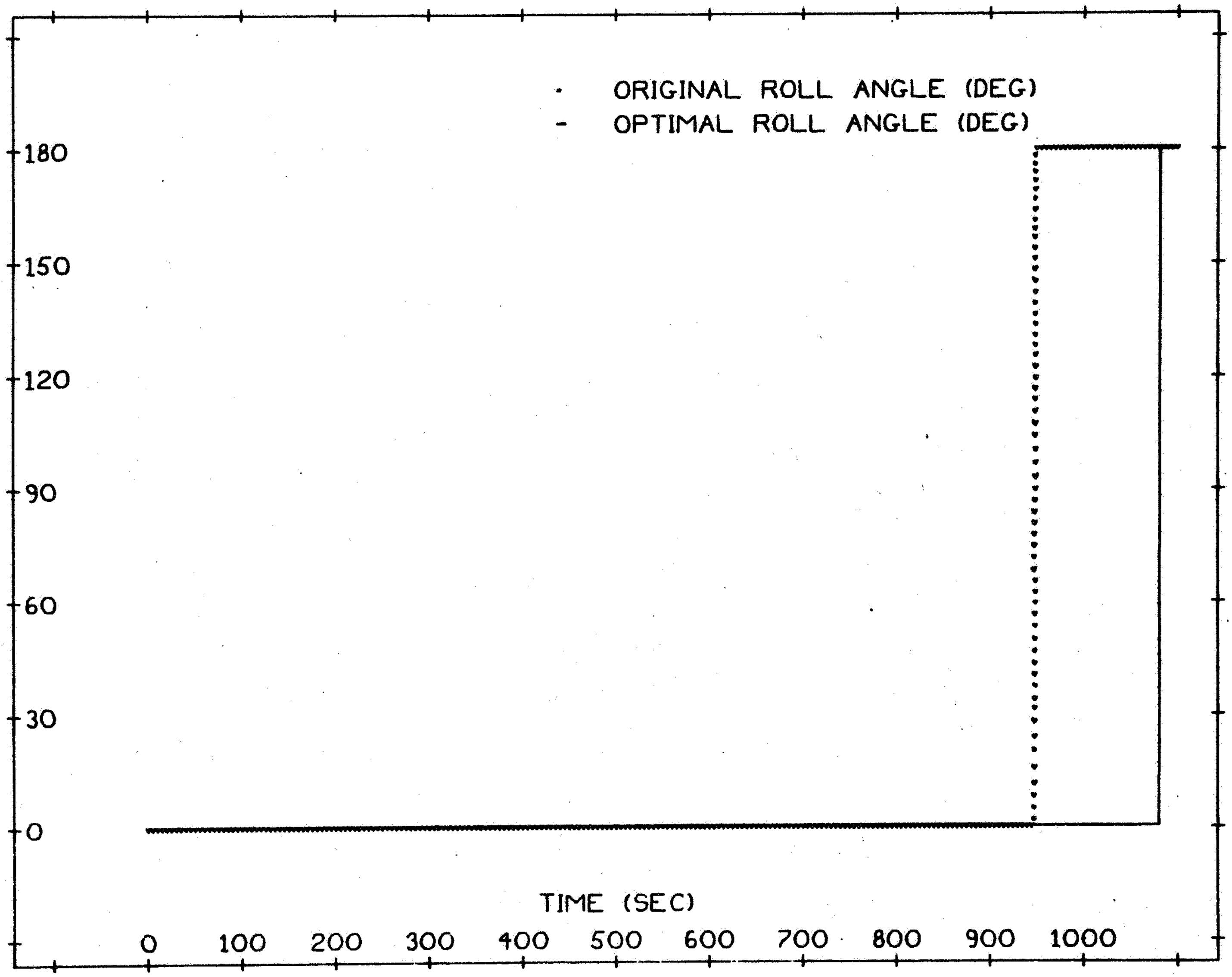


Figure 6. 2. 1b



· ORIGINAL ANGLE OF ATTACK (DEG)
- OPTIMAL ANGLE OF ATTACK (DEG)
60
50
40
30
20
10
0
TIME (SEC)
0
100
200
300
400
500
600
700
800
900
1000

Figure 6.2.1 c



· ORIGINAL ROLL ANGLE (DEG)
- OPTIMAL ROLL ANGLE (DEG)
180
150
120
90
60
30
0
TIME (SEC)
0
100
200
300
400
500
600
700
800
900
1000

Figure 6. 2. 1 d

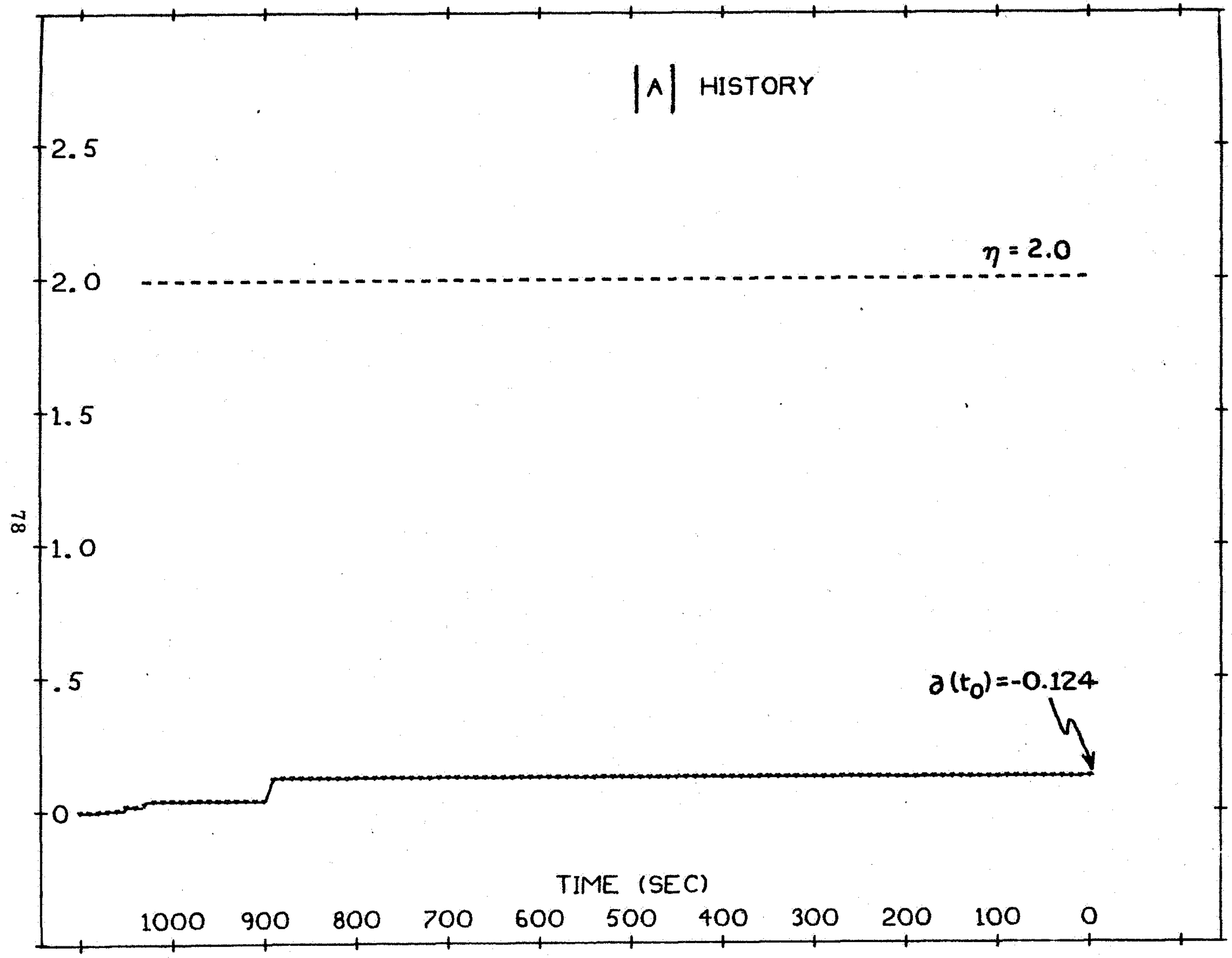

|A| HISTORY
2.5
2.0
1.5
1.0
.5
0
η = 2.0
∂(t0)=-0.124
TIME (SEC)
1000
900
800
700
600
500
400
300
200
100
0

Figure 6.2.1 e

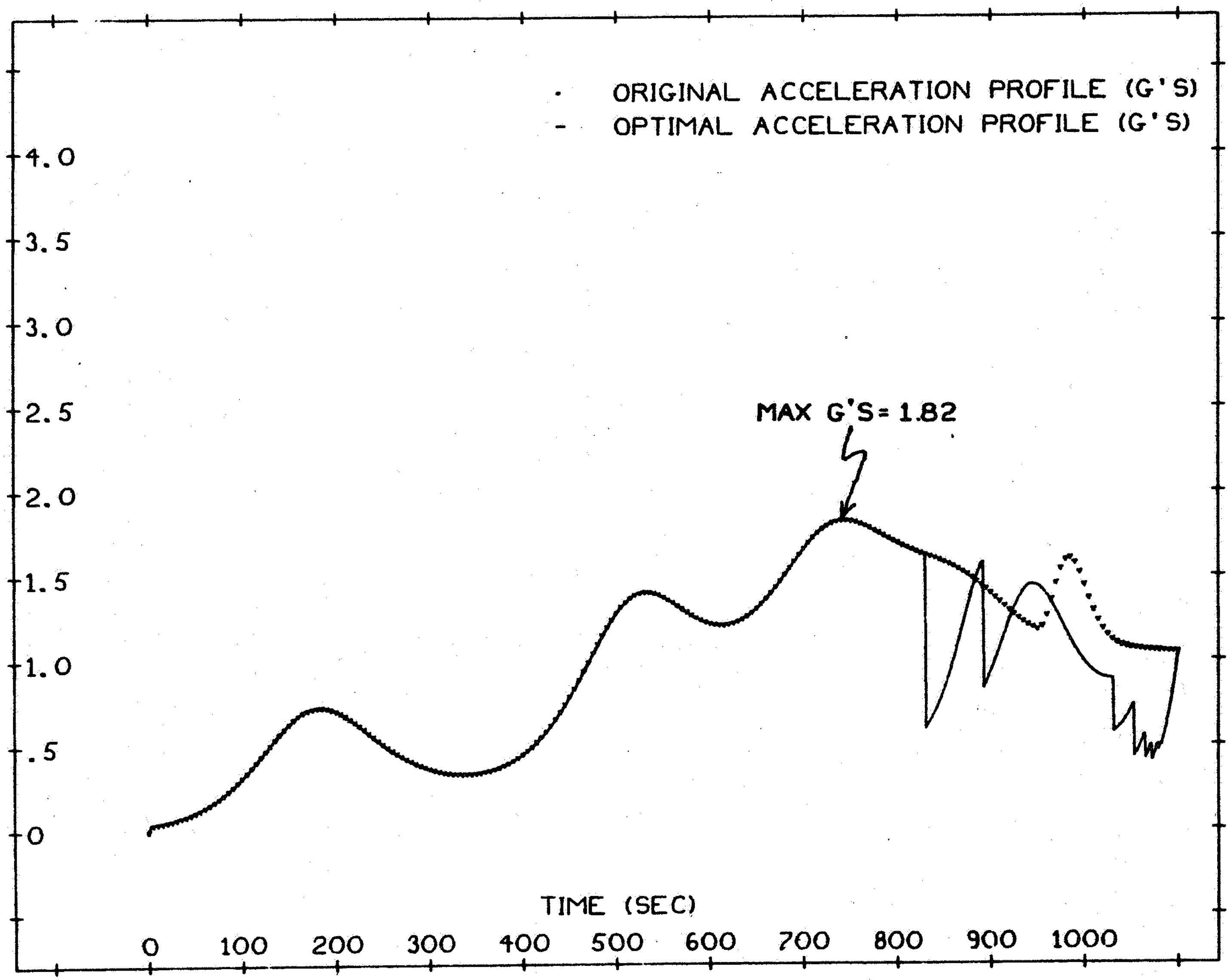
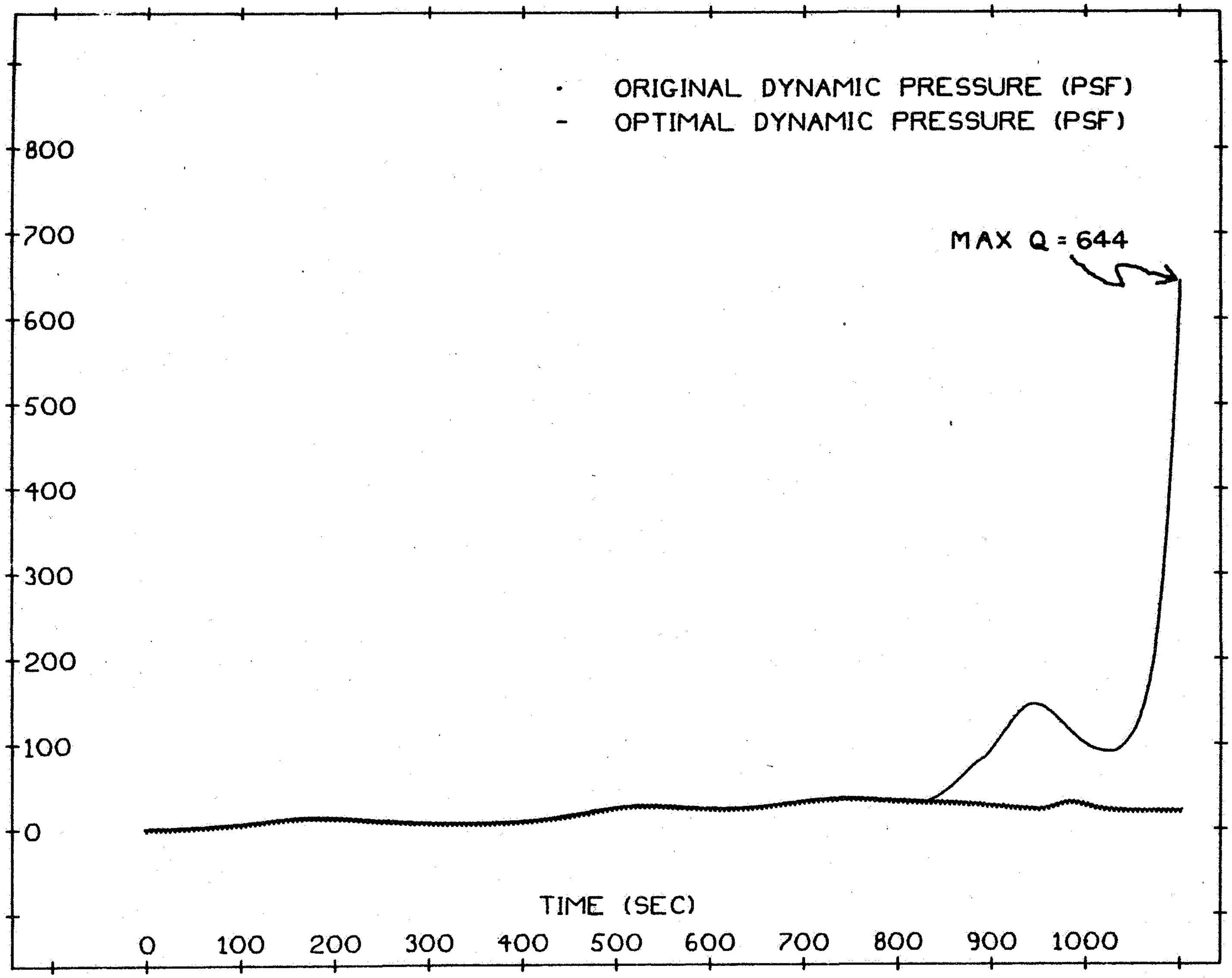

· ORIGINAL ACCELERATION PROFILE (G'S)
- OPTIMAL ACCELERATION PROFILE (G'S)
MAX G'S = 1.82
4.0
3.5
3.0
2.5
2.0
1.5
1.0
.5
0
TIME (SEC)
0
100
200
300
400
500
600
700
800
900
1000

Figure 6.2.1f



ORIGINAL DYNAMIC PRESSURE (PSF)
OPTIMAL DYNAMIC PRESSURE (PSF)
MAX Q = 644
800
700
600
500
400
300
200
100
0
TIME (SEC)
0
100
200
300
400
500
600
700
800
900
1000

Figure 6.2.1 g

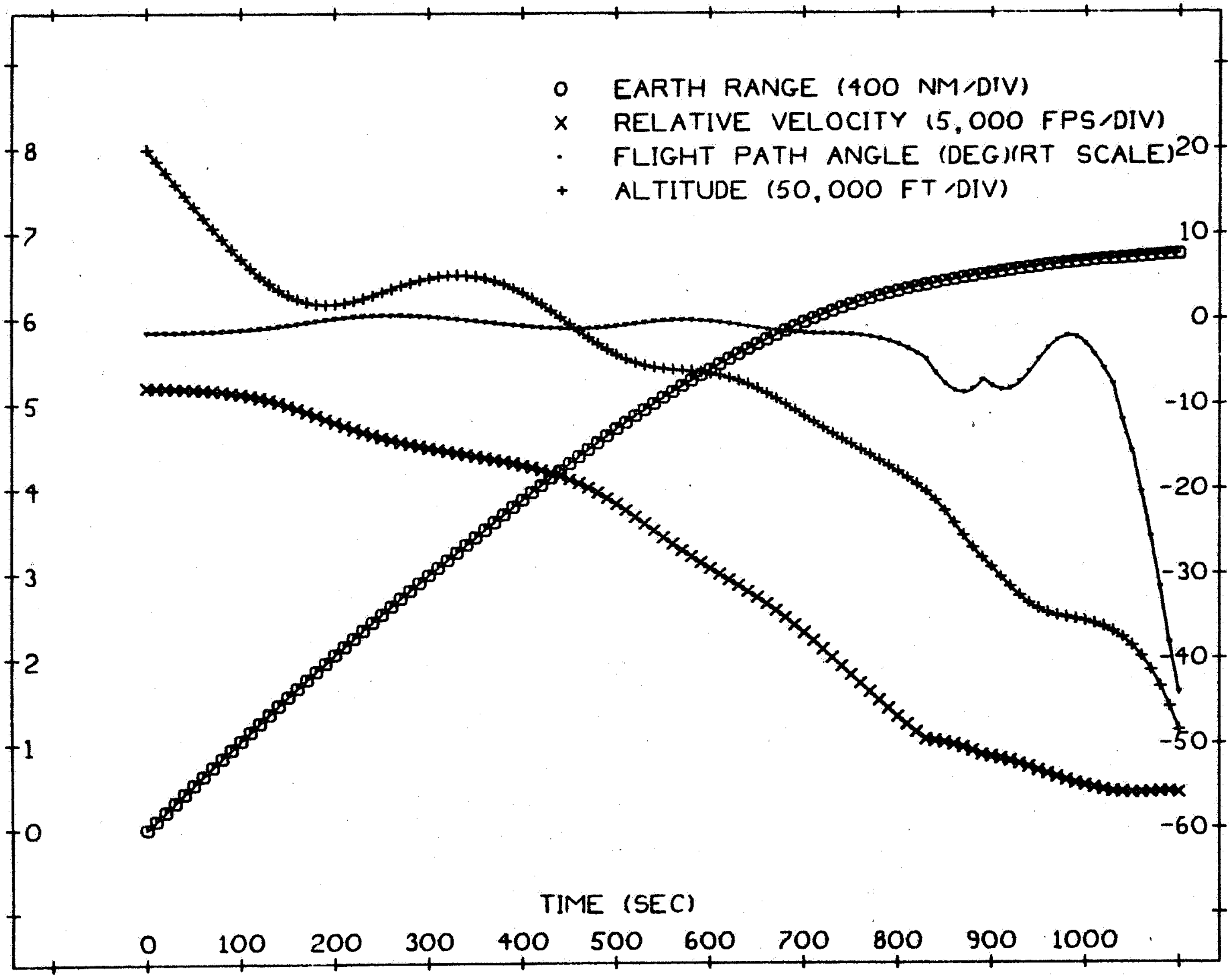

o EARTH RANGE (400 NM/DIV)
x RELATIVE VELOCITY (5,000 FPS/DIV)
· FLIGHT PATH ANGLE (DEG)(RT SCALE)
+ ALTITUDE (50,000 FT/DIV)
TIME (SEC)
0 100 200 300 400 500 600 700 800 900 1000
0 1 2 3 4 5 6 7 8
20 10 0 -10 -20 -30 -40 -50 -60

Figure 6.2.1h

$$t_f = 1100 \text{ sec.} \tag{6.2.9}$$

$$\eta = 2.0 \tag{6.2.10}$$

with a desired down-range of 2700 n.m. Fig. 6.2.1a indicates the decrease in cost with every iteration. It also gives a down-range terminal miss error history with iteration number. Figure 6.2.1b shows the variations of both t_{eff} and t_b with iteration number. Note that the instability of the reverse differential matrix Riccati Eq. (4.3.3) occuring on the initial iteration at t = 212 sec. is eliminated in the iterations thereacter. Figures 6.2.1c and 6.2.1d depict the control histories for angle of attack and roll angle, respectively. The initial nominal and the final optimal control histories are shown for both control variables. Figure 6.2.1e shows the magnitude of the "a" history on the last optimizing iteration, when its absolute value never exceeds η on any portion of the time interval of interest. On preceeding iterations, t_{eff} is determined by the time at which $|a|$ first exceeds η, and on subsequent iterations, more and more of the $|a|$ history falls below the $|a| = \eta$ value. For the continuous case, $|a|$ should identically vanish, for it is a measure of how close the nominal control is to the optimal control, as indicated in Eq. (4.3.1). However, where finite size time steps are employed, a non-zero tolerance value of η is utilized. The values of η employed in these cases are rather large, due to the long time step durations and the lengthy overall time interval of interest. In this first case, the final $|a|$ history falls well below the η value. Figures 6.2.1f - 6.2.1g depict initial and final optimal acceleration and dynamic pressure histories. Here dynamic pressure increased markedly and is essentially irrelevant to the first case, for $W_q = 0$. Note the decrease in g's after t = 820 sec., for all accelerations above zero were penalized. The kinks or jumps in the acceleration profiles are due to the finite jumps in the angle of attack history. Optimal state variable histories are finally shown in Fig. 6.2.1h.

In an effort to eliminate the instability point, the time interval of interest was shortened, leading to a new set of initial conditions:

$$x_r(t_0) = 1699.4 \text{ n.m.} \tag{6.2.11}$$

$$v(t_0) = 20{,}736.13 \text{ fps} \tag{6.2.12}$$

$$\gamma(t_0) = -1.11203 \text{ deg.} \tag{6.2.13}$$

$$h(t_0) = 300{,}061.4 \text{ ft.} \tag{6.2.14}$$

$$t_f = 657 \text{ sec.} \tag{6.2.15}$$

These values were extracted from an open loop trajectory with the original initial conditions of Eqs. (6.2.1) - (6.2.4), and maintaining constant control histories of $\phi = 0^{\circ}$ and $\alpha = 60^{\circ}$ down to about 300,000 ft. in altitude, thus producing an essentially "equivalent" set of initial boundary conditions. The following two cases show the effectiveness of shortening the total time interval of interest.

Figures 6.2.2a - 6.2.2h present a case identical to the first except for the following differences:

$$\sigma_q = 100 \tag{6.2.16}$$

or

$$Wq = .0001 \tag{6.2.17}$$

$$W_{g's} = 0 \tag{6.2.18}$$

$$\eta = 2.5 \tag{6.2.19}$$

The new initial conditions for the shortened trajectory are employed. Now dynamic pressure is penalized, while the acceleration history is essentially ignored. Figure 6.2.2a shows again the decrease in cost

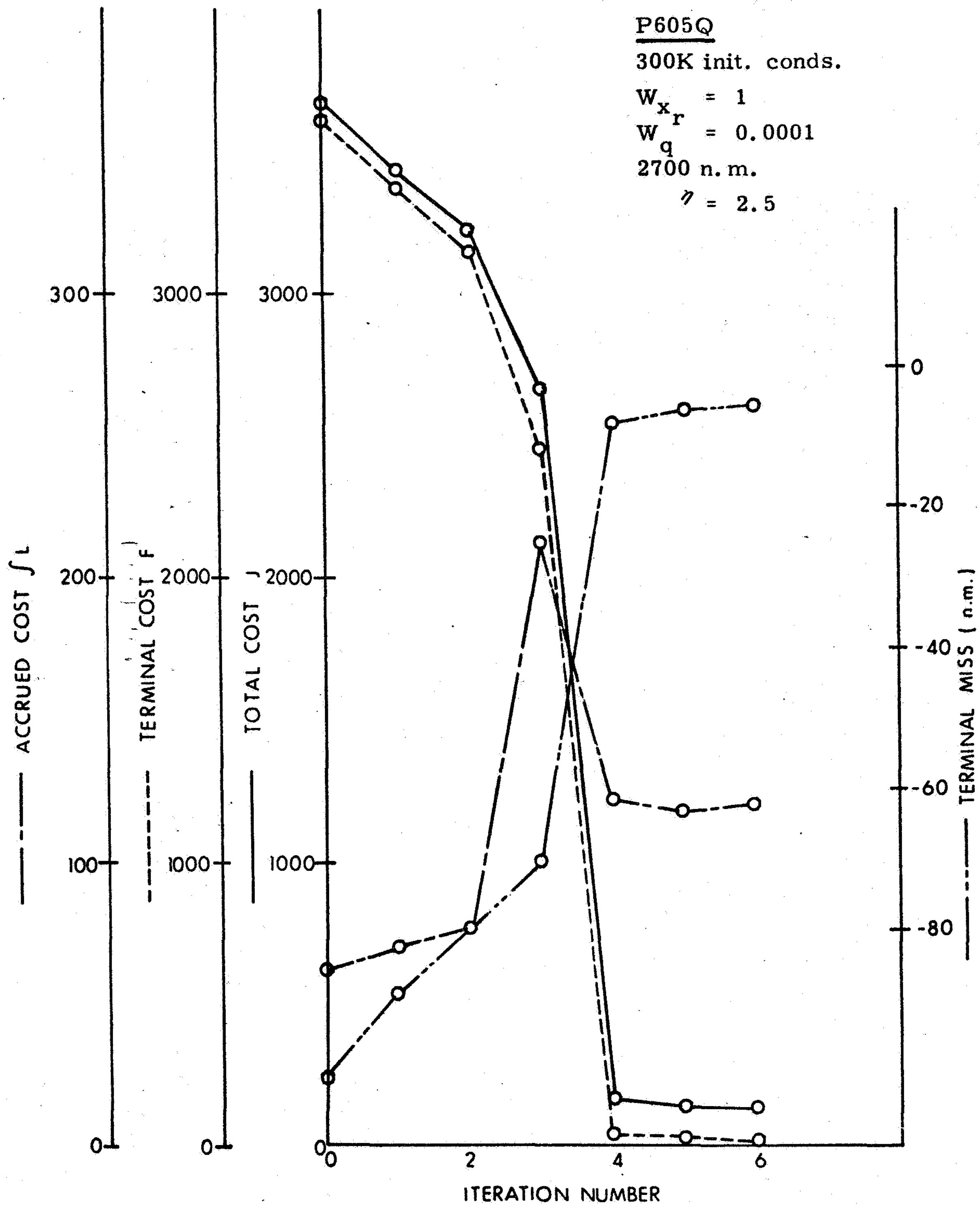

P605Q
300K init. conds.
W_{x_r} = 1
W_q = 0.0001
2700 n.m.
η = 2.5
ACCRUED COST $\int L$
TERMINAL COST F
TOTAL COST J
TERMINAL MISS (n.m.)
300
200
100
0
3000
2000
1000
0
3000
2000
1000
0
0
-20
-40
-60
-80
0
2
4
6
ITERATION NUMBER

Figure 6.2.2a

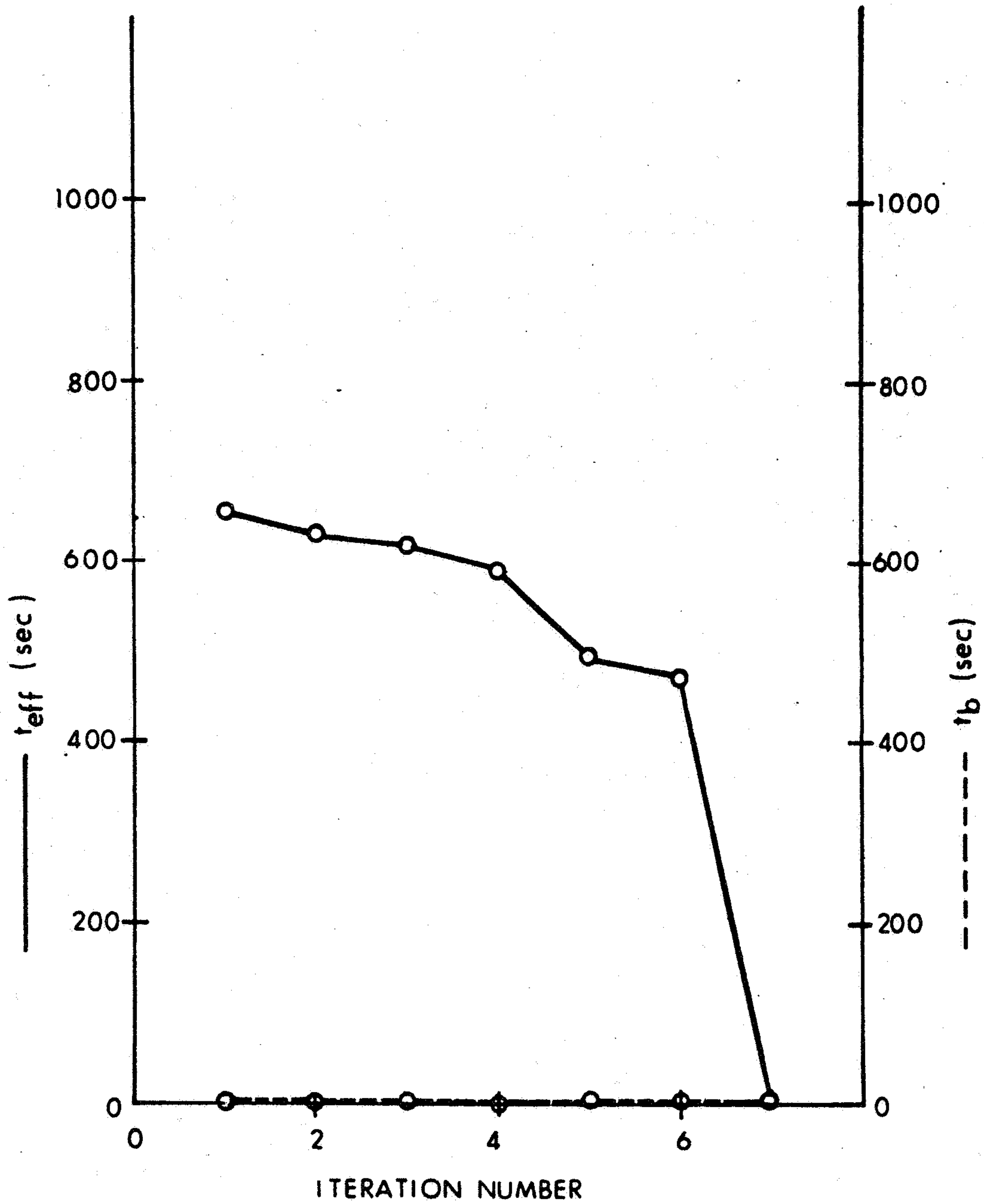


Figure 6. 2. 2b

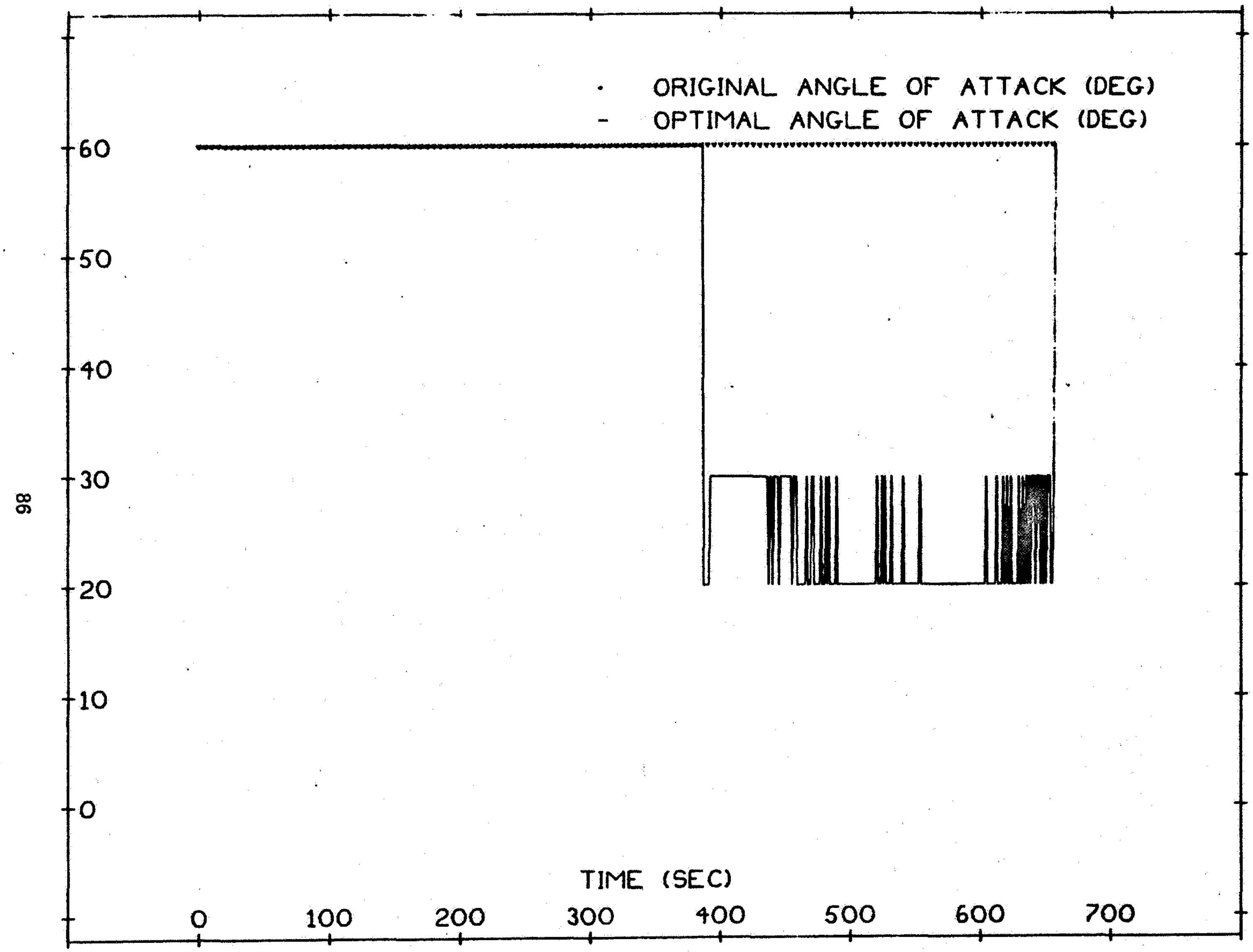
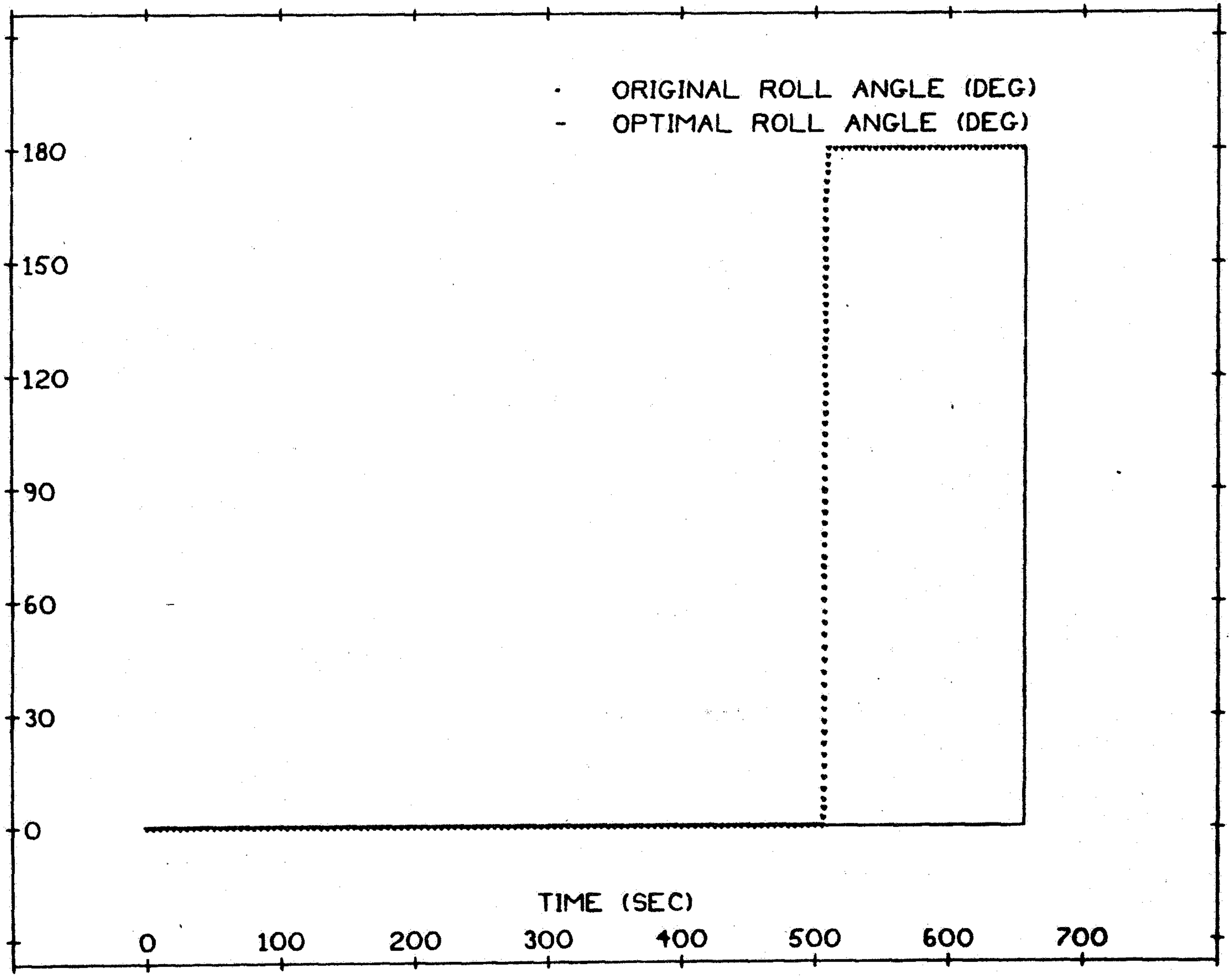

· ORIGINAL ANGLE OF ATTACK (DEG)
- OPTIMAL ANGLE OF ATTACK (DEG)
60
50
40
30
20
10
0
TIME (SEC)
0
100
200
300
400
500
600
700

Figure 6.2.2c



· ORIGINAL ROLL ANGLE (DEG)
- OPTIMAL ROLL ANGLE (DEG)
180
150
120
90
60
30
0
TIME (SEC)
0
100
200
300
400
500
600
700

Figure 6.2.2d

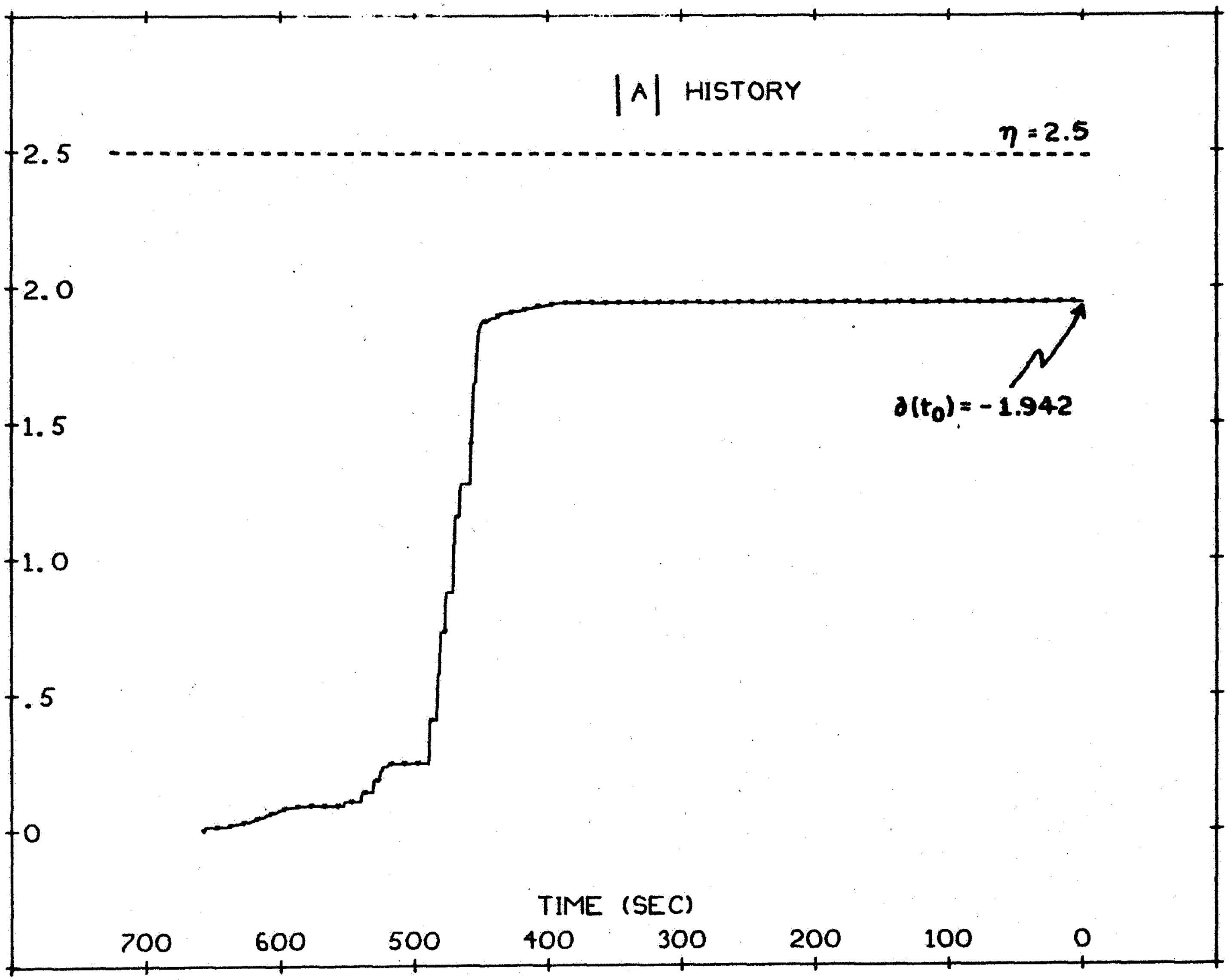
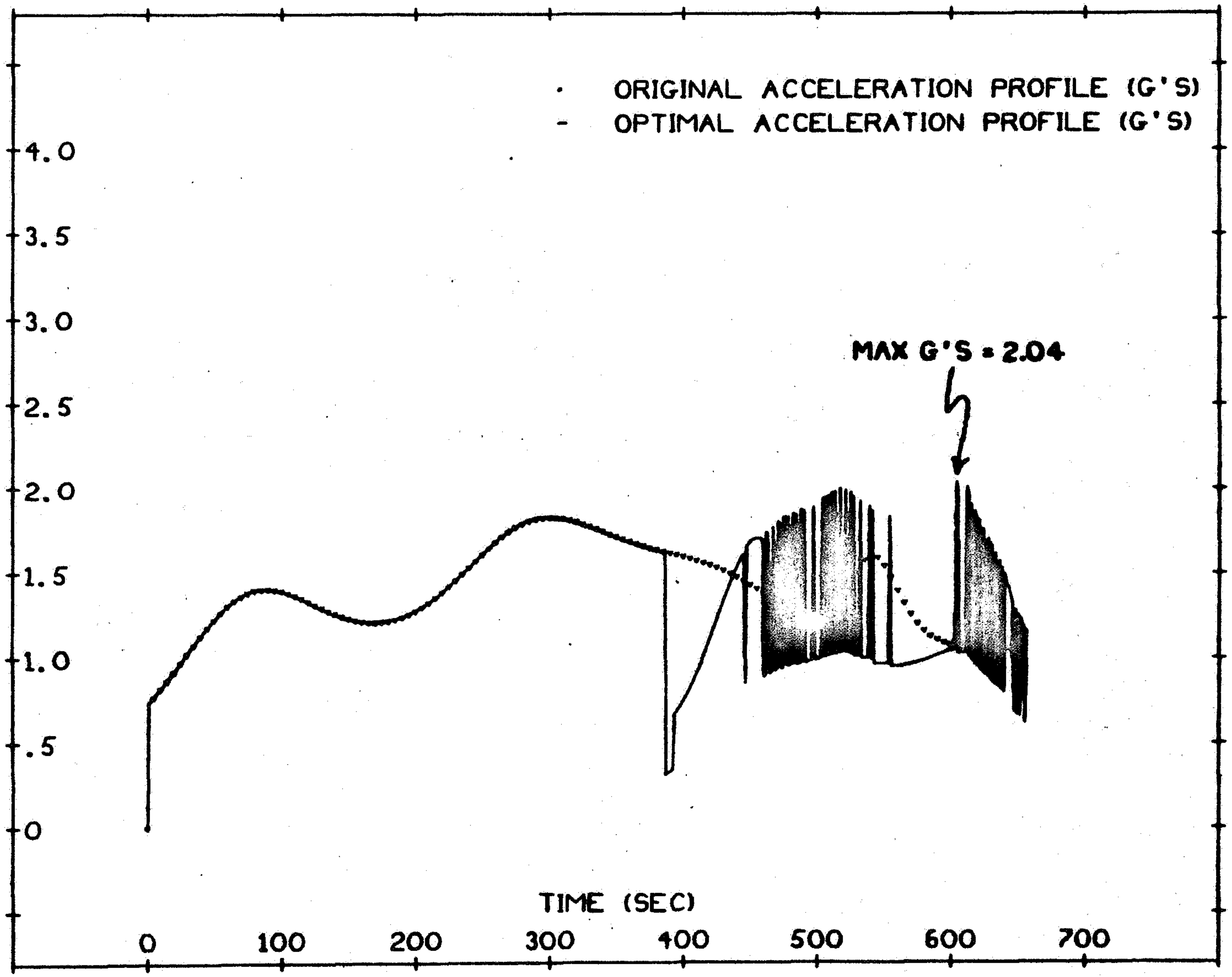
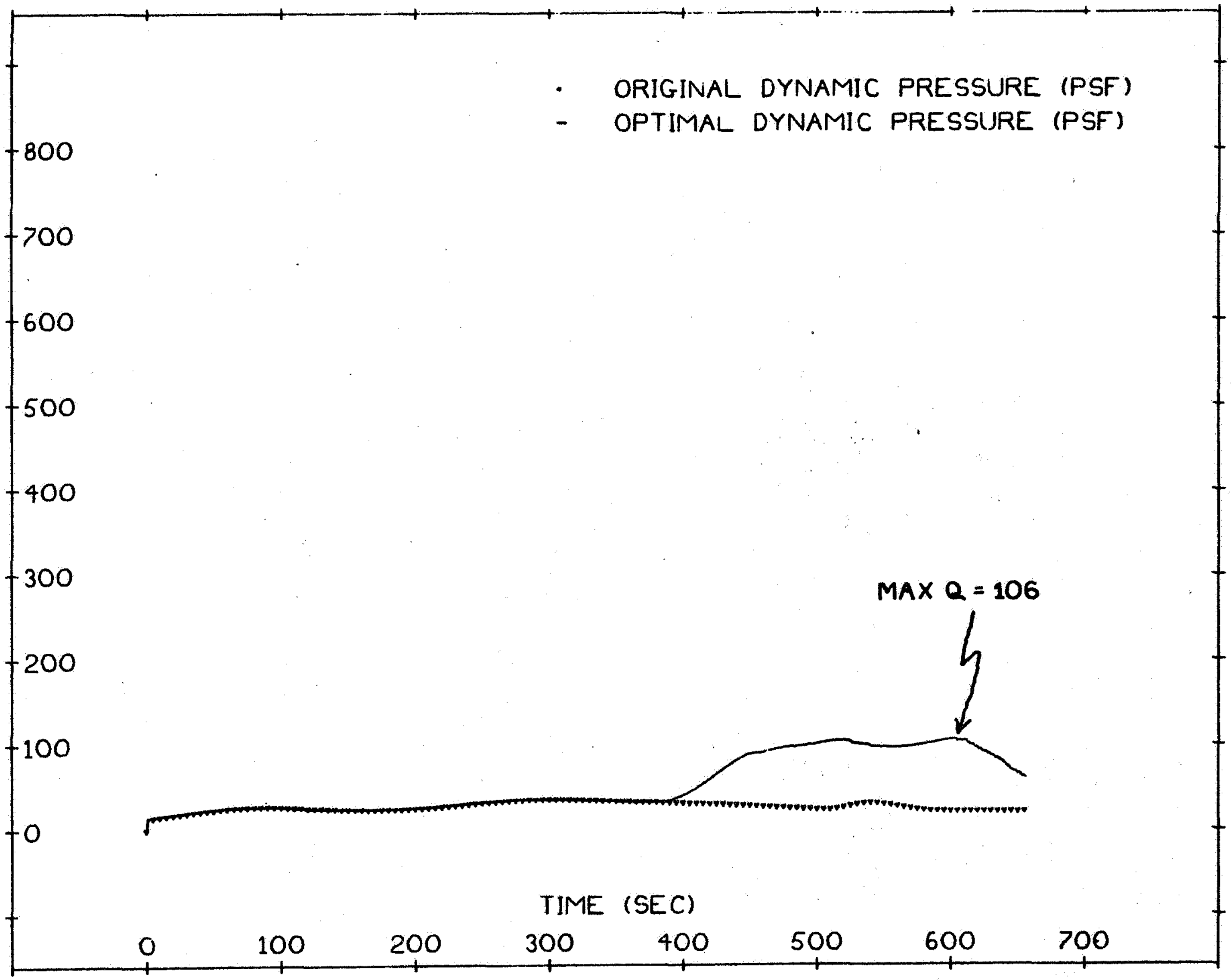

|A| HISTORY
η = 2.5
∂(t0) = -1.942
2.5
2.0
1.5
1.0
.5
0
TIME (SEC)
700
600
500
400
300
200
100
0

Figure 6.2.2e



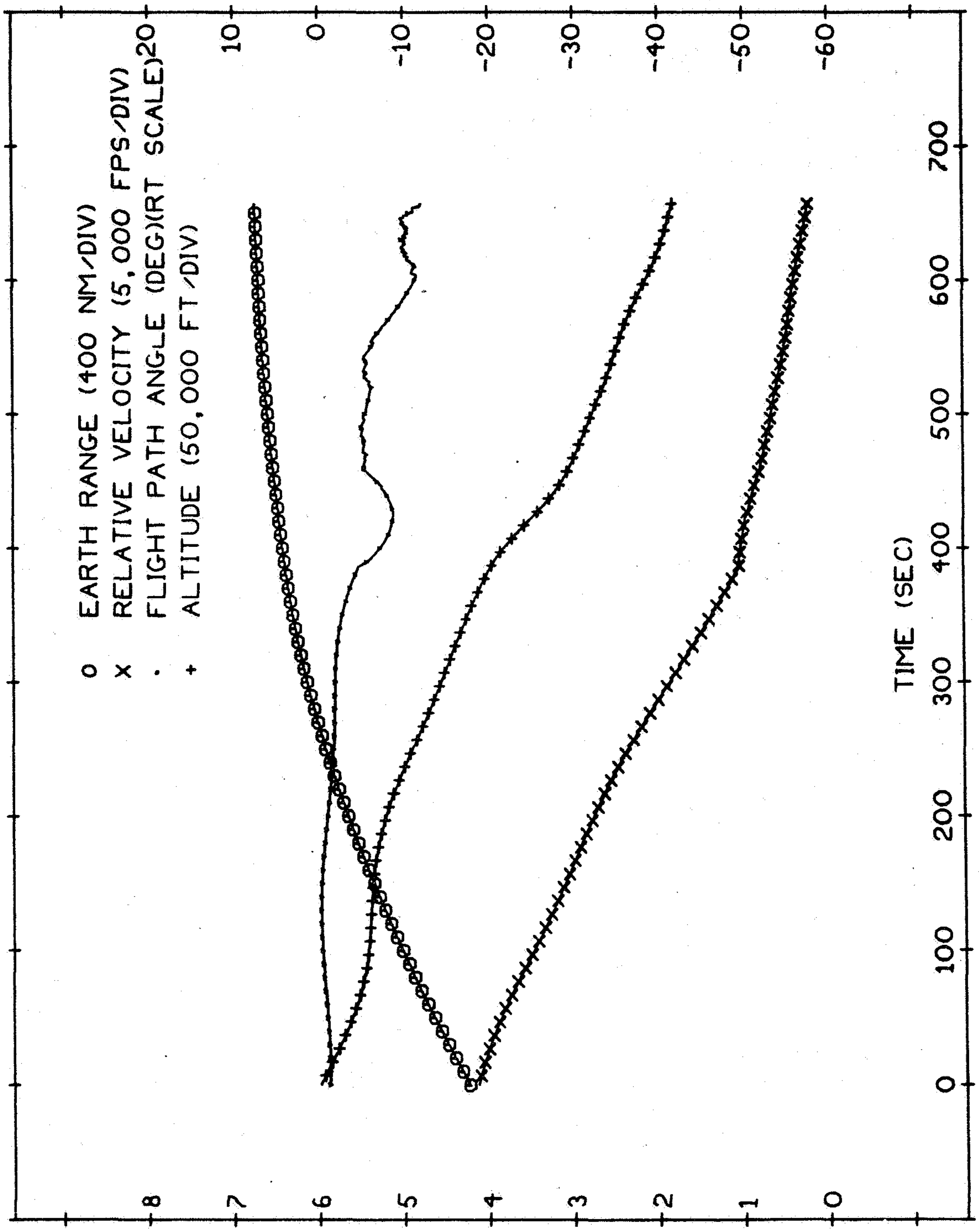
ORIGINAL ACCELERATION PROFILE (G'S)
OPTIMAL ACCELERATION PROFILE (G'S)
MAX G'S = 2.04
4.0
3.5
3.0
2.5
2.0
1.5
1.0
.5
0
TIME (SEC)
0
100
200
300
400
500
600
700

Figure 6.2.2f



· ORIGINAL DYNAMIC PRESSURE (PSF)
- OPTIMAL DYNAMIC PRESSURE (PSF)
MAX Q = 106
800
700
600
500
400
300
200
100
0
TIME (SEC)
0
100
200
300
400
500
600
700

Figure 6.2.2 g



o EARTH RANGE (400 NM/DIV)
x RELATIVE VELOCITY (5,000 FPS/DIV)
. FLIGHT PATH ANGLE (DEG)(RT SCALE)
+ ALTITUDE (50,000 FT/DIV)
TIME (SEC)
0
100
200
300
400
500
600
700
0
1
2
3
4
5
6
7
8
20
10
0
-10
-20
-30
-40
-50
-60

Figure 6.2.2h

with every iteration, while Fig. 6.2.2b evidences the complete elimination of an instability point, and thus $t_b = 0$ at every iteration. Figure 6.2.2c and 6.2.2d indicate the initial nominal and final optimal control histories, and Fig. 6.2.2e gives the $|a|$ variation with time. In Fig. 6.2.2f, larger accelerations for longer periods of time are allowed than in Fig. 6.2.1f, since the gravity profile is now given zero weighting. Comparing Fig. 6.2.2g with 6.2.1g, a cost improvement in dynamic pressure is observed, as was intended by the inclusion of the weighting factor, W_q, for this particular case. The undesirable increase in dynamic pressure of the initial nominal history as compared with the final optimal history, and hence an increase in the dynamic pressure cost term, is more than offset by the cost improvement in attainment of the terminal desired down range, which was more heavily weighted in Eq. (6.2.8). This terminal miss distance in down-range is given in Fig. 6.2.2a as a function of iteration number. Thus, the importance of the relative ratios of the weightings is brought out, and the consequences of a soft constraint quadratic penalty cost approach are made known.
The decrease in total cost at every iteration is the determining factor in the optimization procedure, and the relative importances of the various penalties are reflected by the magnitude and more importantly by the ratios of the various weighting factors. The optimal state histories are displayed in Fig. 6.2.2h.

The following weighting ratios lead to the results depicted in Figs. 6.2.3a - 6.2.3h:

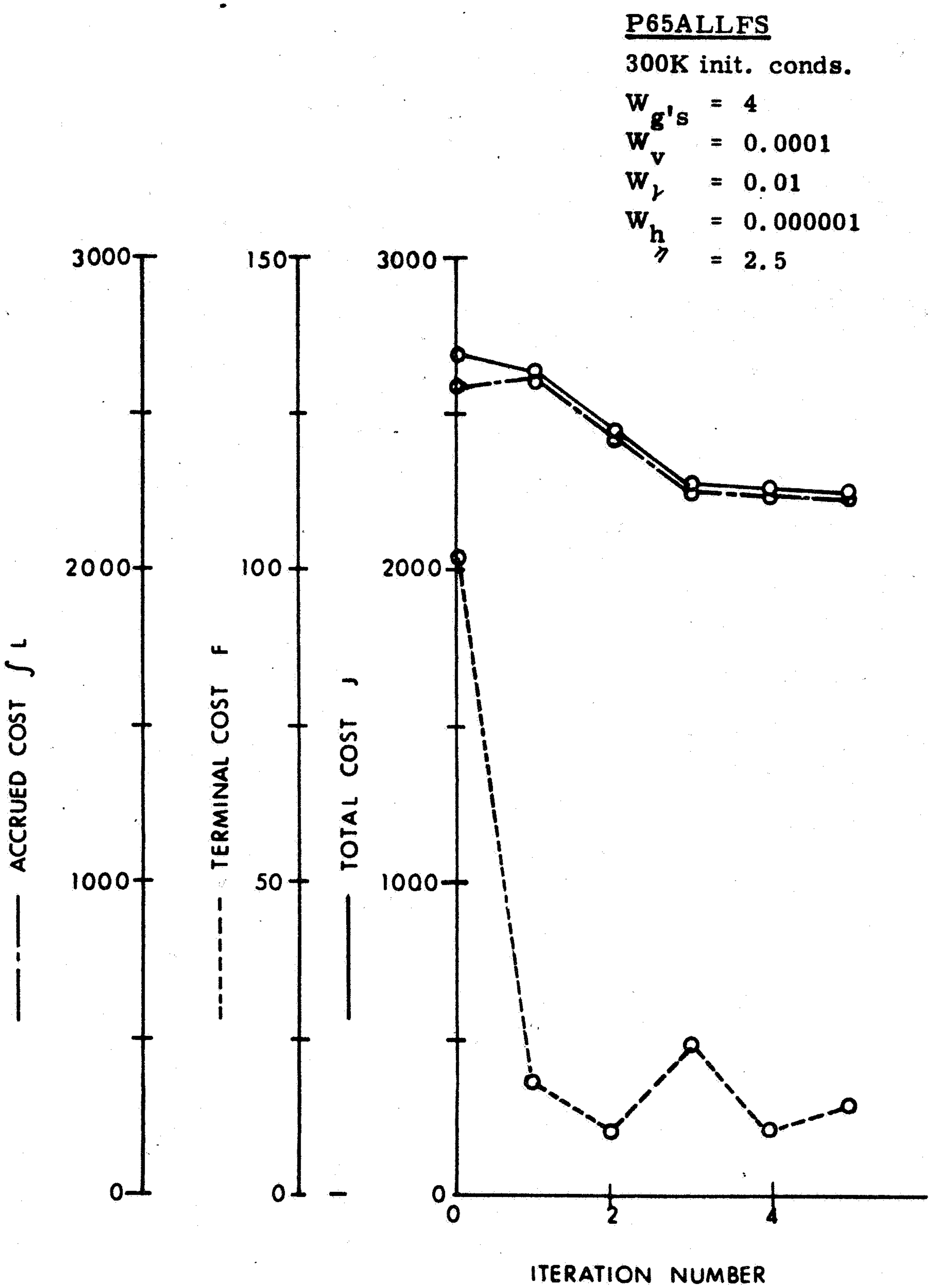
$$W_{x_r} = 0 \tag{6.2.20}$$

$$W_{g's} = 4 \tag{6.2.21}$$

$$W_v = .0001 \tag{6.2.22}$$

$$W_\gamma = .01 \tag{6.2.23}$$

$$W_h = .000001 \tag{6.2.24}$$



P65ALLFS
300K init. conds.
$W_{g's}$ = 4
W_v = 0.0001
W_γ = 0.01
W_h = 0.000001
η = 2.5
3000
2000
1000
0
150
100
50
0
ACCRUED COST $\int L$
TERMINAL COST F
TOTAL COST J
0
2
4
ITERATION NUMBER

Figure 6.2.3a

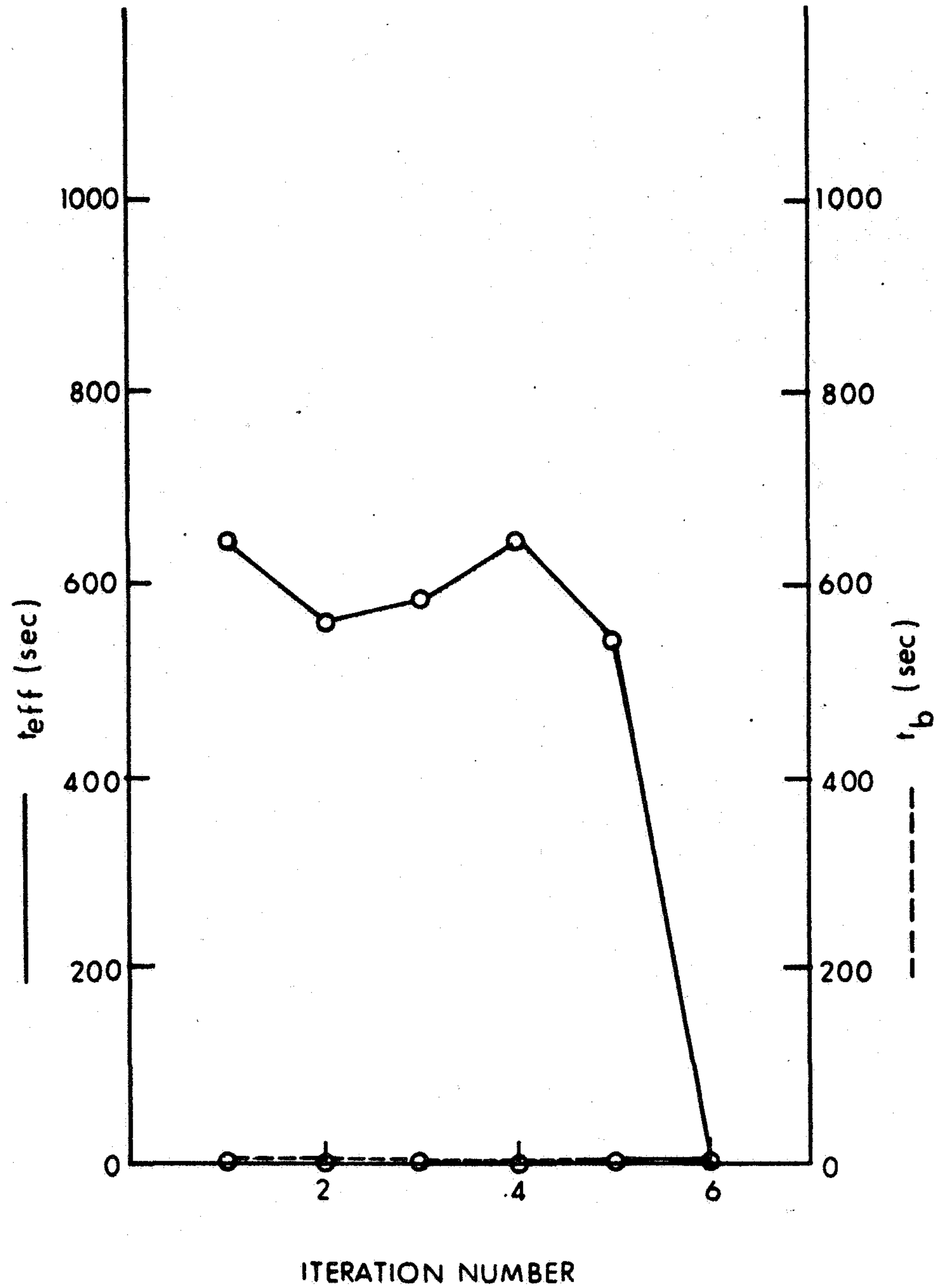
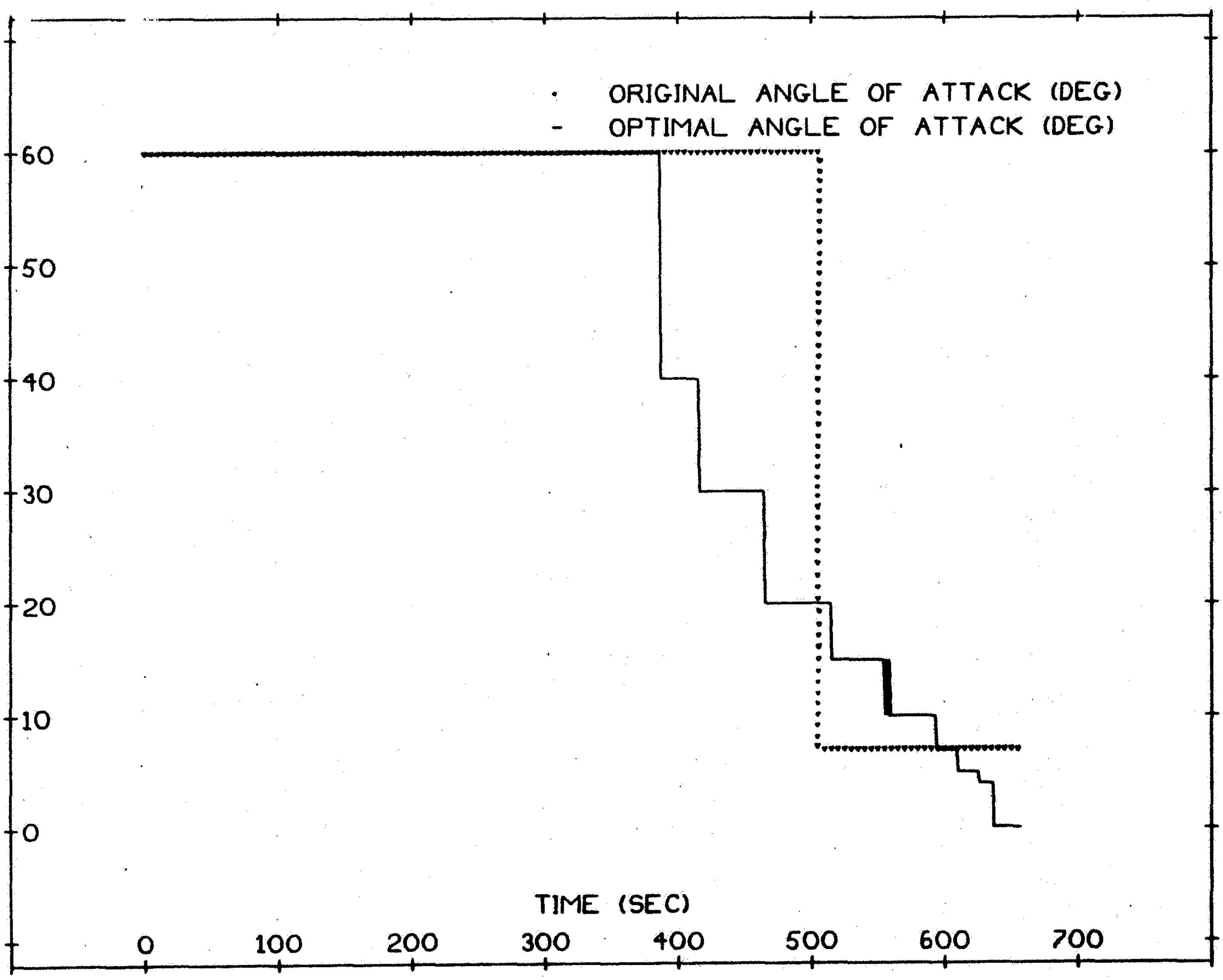
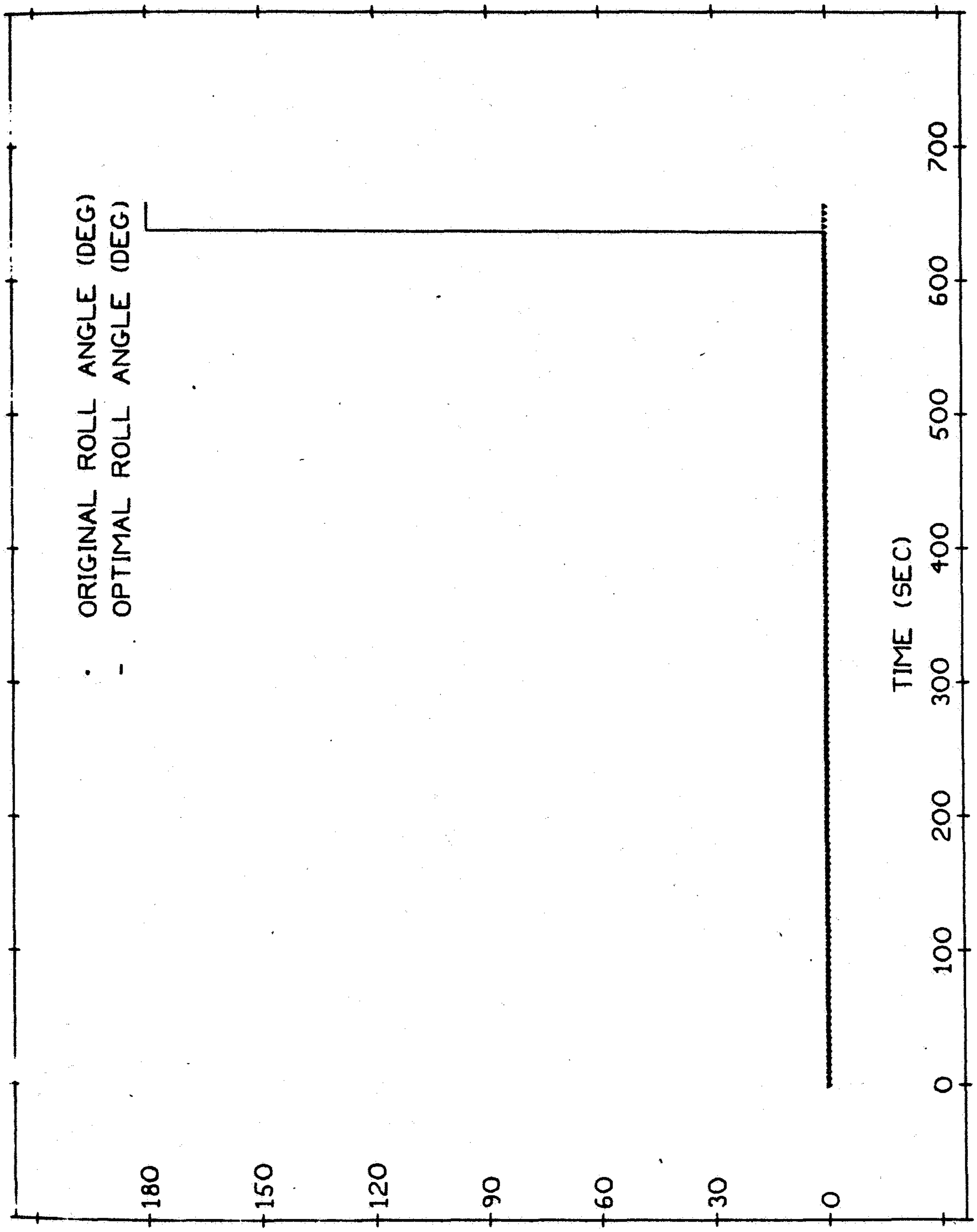


Figure 6. 2. 3b



· ORIGINAL ANGLE OF ATTACK (DEG)
- OPTIMAL ANGLE OF ATTACK (DEG)
60
50
40
30
20
10
0
TIME (SEC)
0
100
200
300
400
500
600
700

Figure 6.2.3c



. ORIGINAL ROLL ANGLE (DEG)
- OPTIMAL ROLL ANGLE (DEG)
180
150
120
90
60
30
0
0
100
200
300
400
500
600
700
TIME (SEC)

Figure 6.2.3d

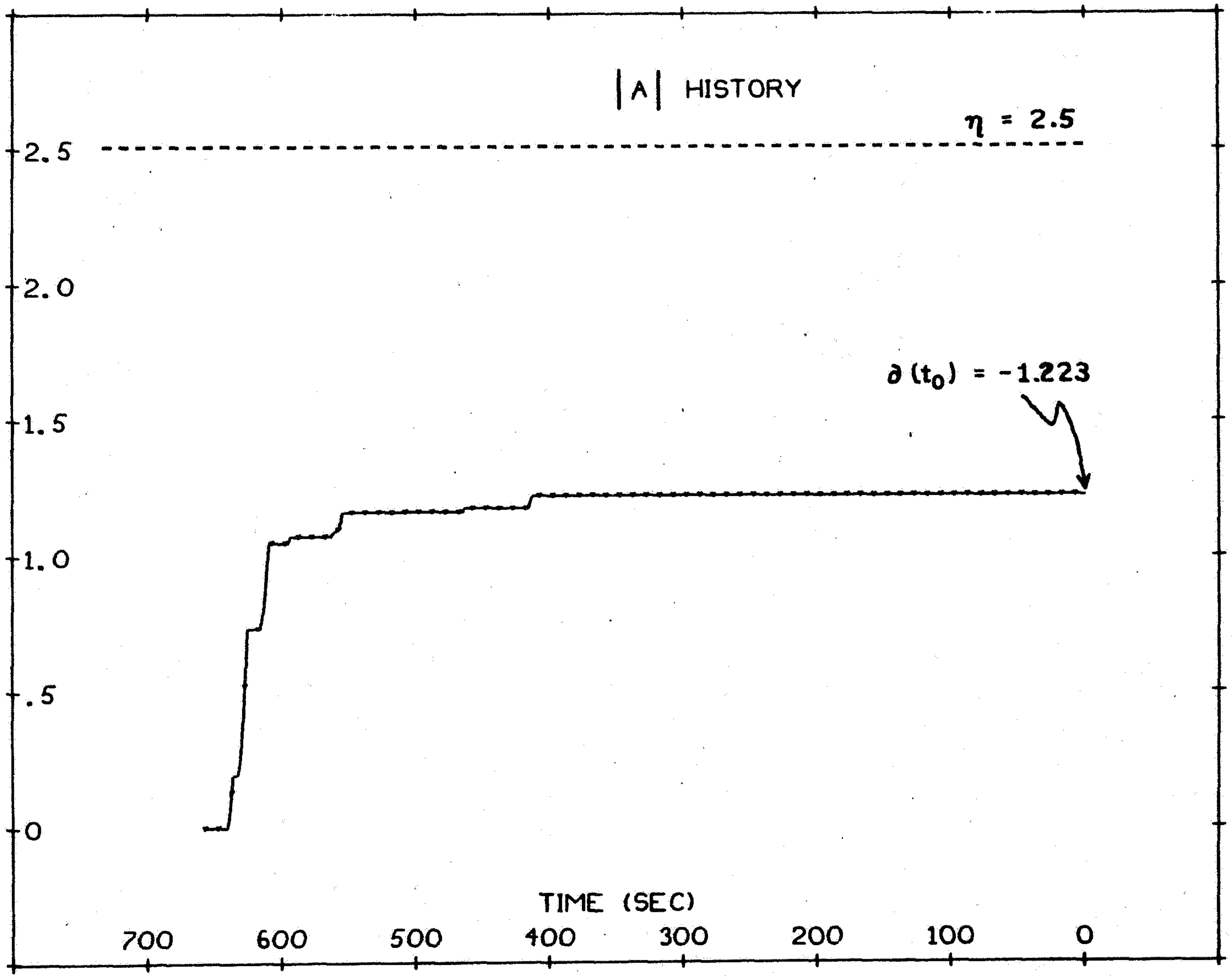
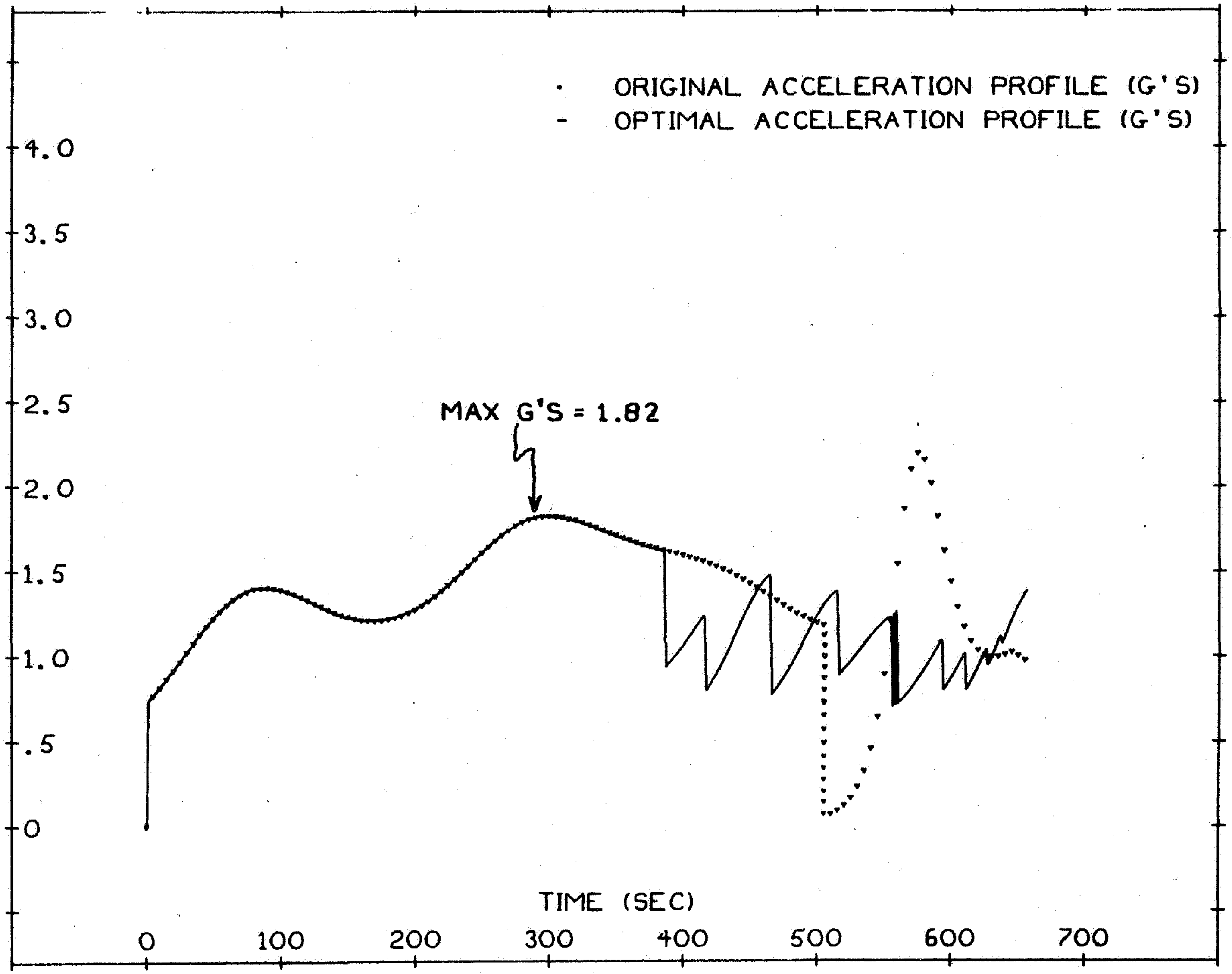
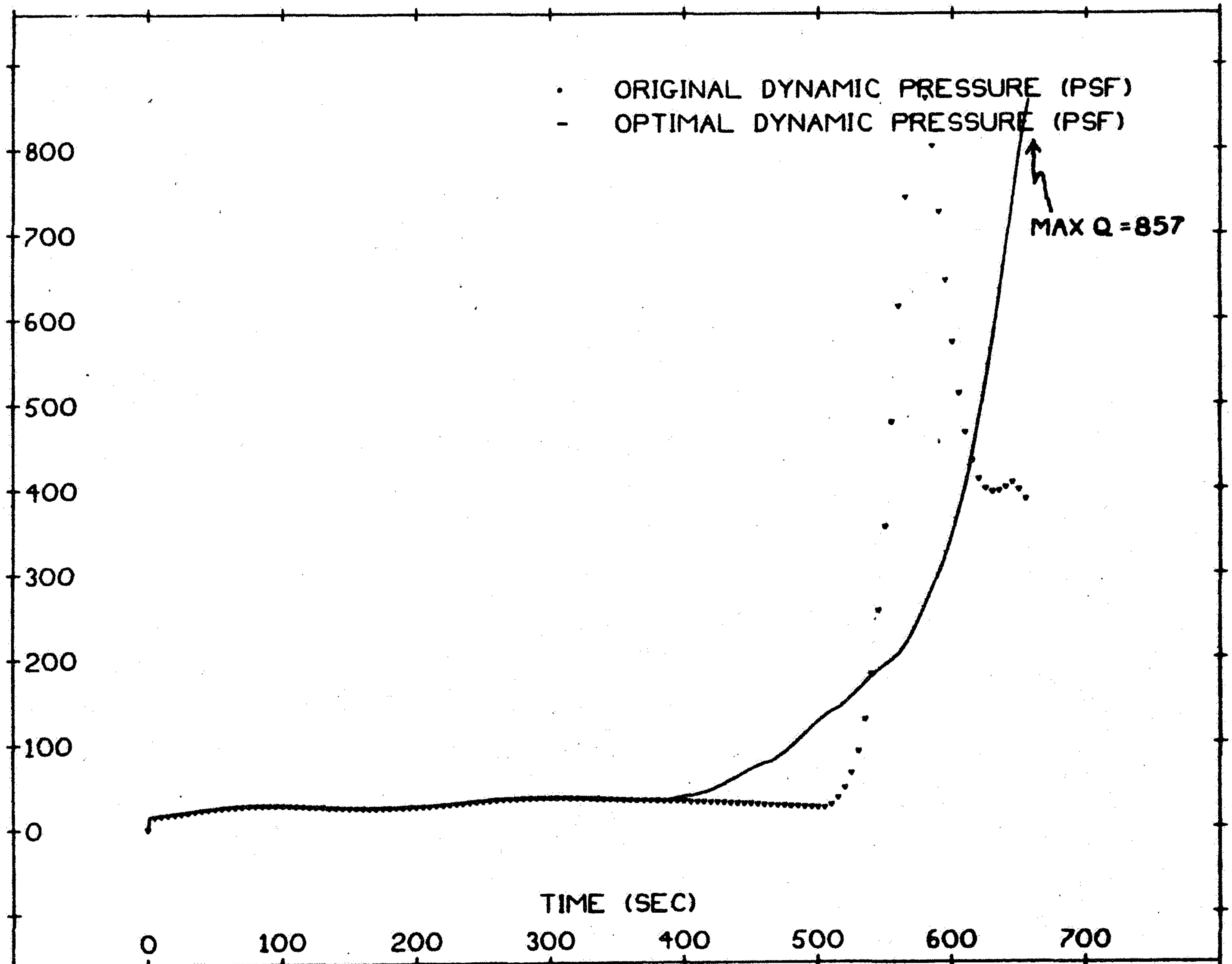

|A| HISTORY
η = 2.5
∂ (t0) = -1.223
2.5
2.0
1.5
1.0
.5
0
TIME (SEC)
700
600
500
400
300
200
100
0

Figure 6.2.3e



· ORIGINAL ACCELERATION PROFILE (G'S)
- OPTIMAL ACCELERATION PROFILE (G'S)
MAX G'S = 1.82
4.0
3.5
3.0
2.5
2.0
1.5
1.0
.5
0
TIME (SEC)
0
100
200
300
400
500
600
700

Figure 6.2.3f



ORIGINAL DYNAMIC PRESSURE (PSF)
OPTIMAL DYNAMIC PRESSURE (PSF)
MAX Q = 857
800
700
600
500
400
300
200
100
0
TIME (SEC)
0
100
200
300
400
500
600
700

Figure 6. 2. 3g

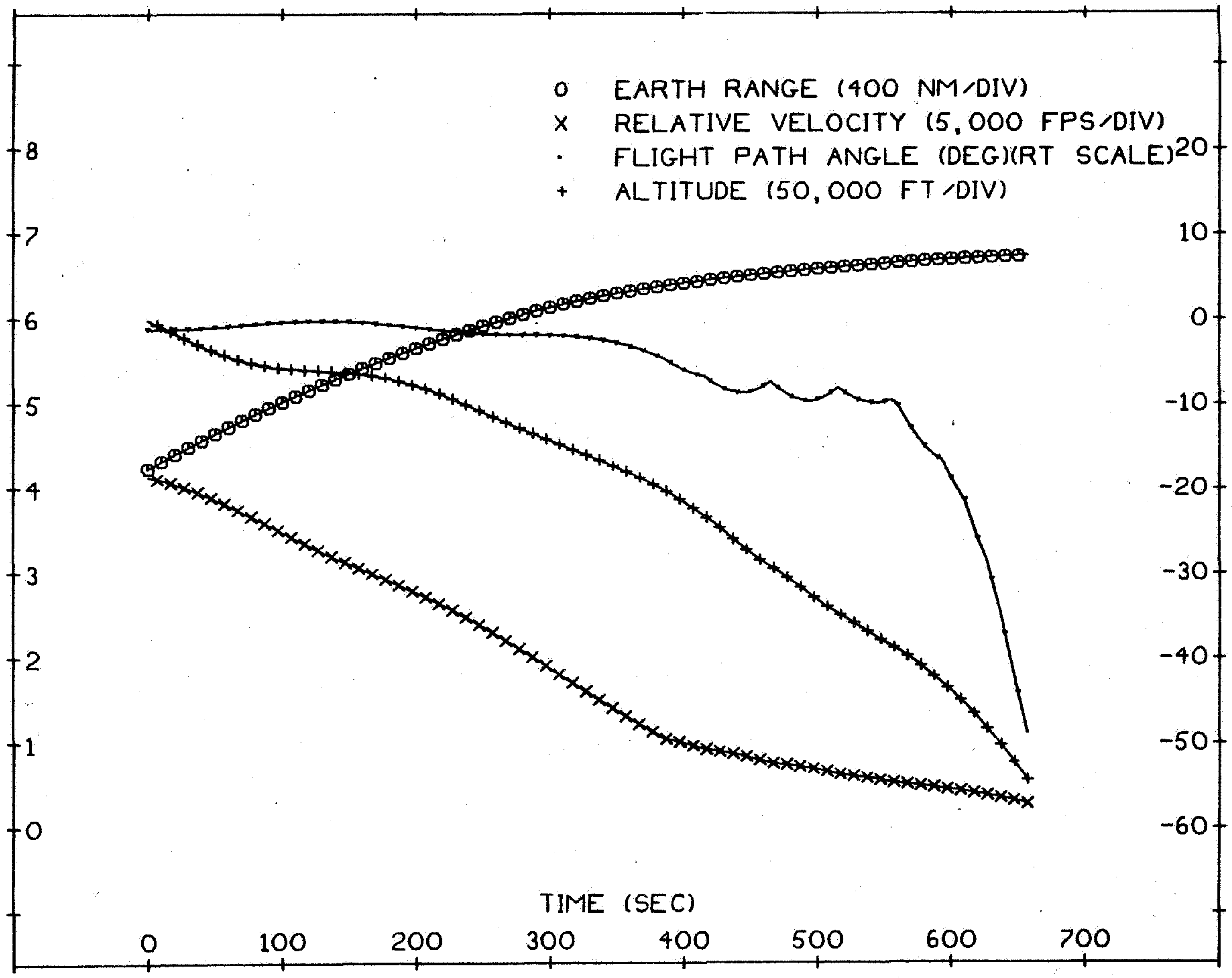

o EARTH RANGE (400 NM/DIV)
x RELATIVE VELOCITY (5,000 FPS/DIV)
· FLIGHT PATH ANGLE (DEG)(RT SCALE)
+ ALTITUDE (50,000 FT/DIV)
TIME (SEC)
0
100
200
300
400
500
600
700
8
7
6
5
4
3
2
1
0
20
10
0
-10
-20
-30
-40
-50
-60

Figure 6. 2. 3 h

The desired terminal state values are a velocity between 100 and 1000 fps, a flight path angle between -10 and 10 deg., and an altitude between 10,000 and 70,000 ft. These soft constraints are much less heavily weighted than the acceleration. Figure 6.2.3a indicates the decrease in total cost at every iteration step. The overall decrease in terminal cost is also indicated, but is much smaller than the decreases in the accrued cost for the acceleration profile. Figure 6.2.3b shows the successful elimination of the instability point in the reverse differential matrix Riccati equation. Figure 6.2.3c and 6.2.3d present the initial nominal and final optimal control histories, and Fig. 6.2.3e the $|a|$ variation with time. Figures 6.2.3f and 6.2.3g reveal the acceleration and dynamic pressure profiles with the former penalized and the latter essentially ignored. Final optimal state histories are exhibited in Fig. 6.2.3h.

The fourth case is the same as the third with the following changes:

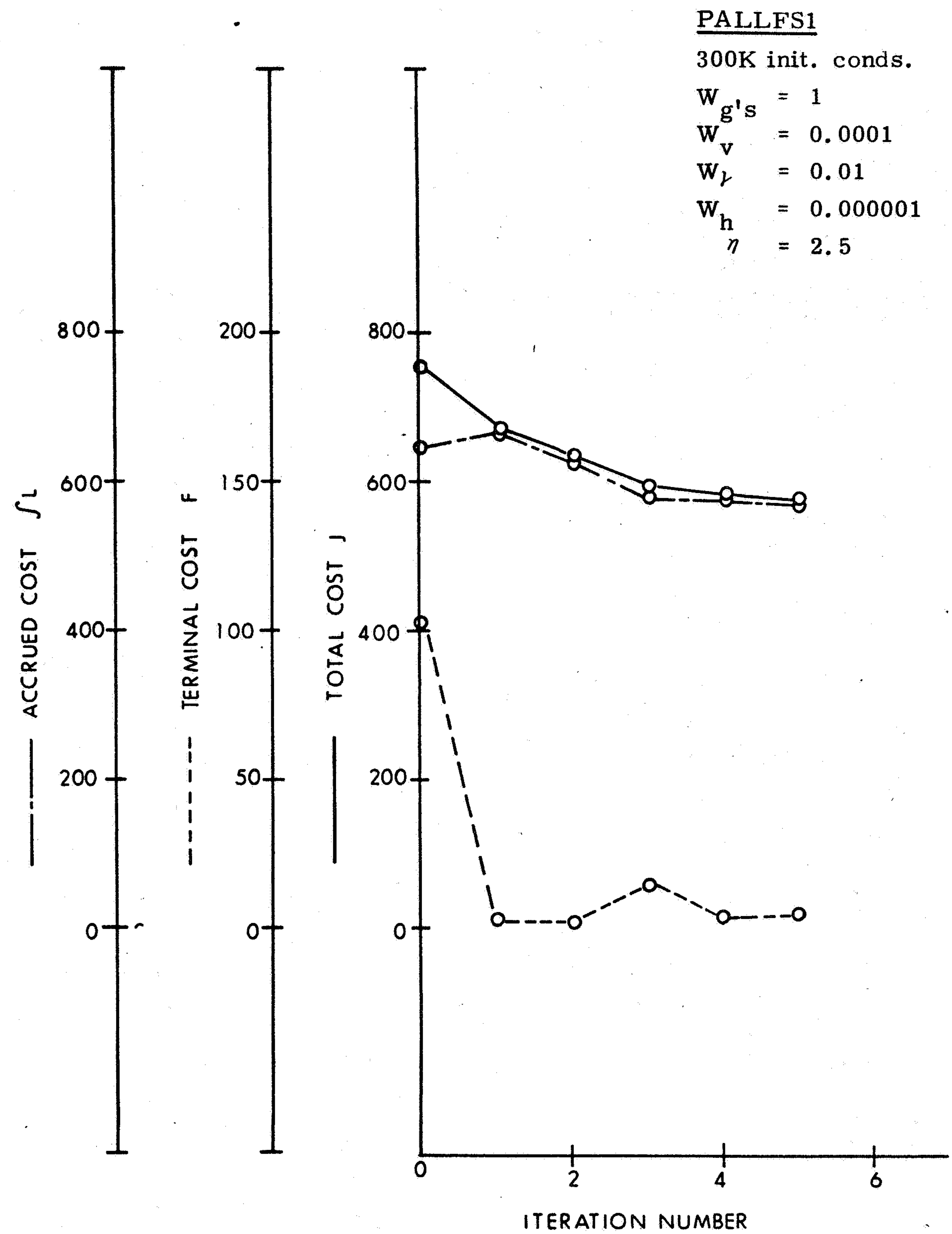
$$\sigma_{g's} = 1 \qquad (6.2.25)$$

or

$$W_{g's} = 1 \qquad (6.2.26)$$

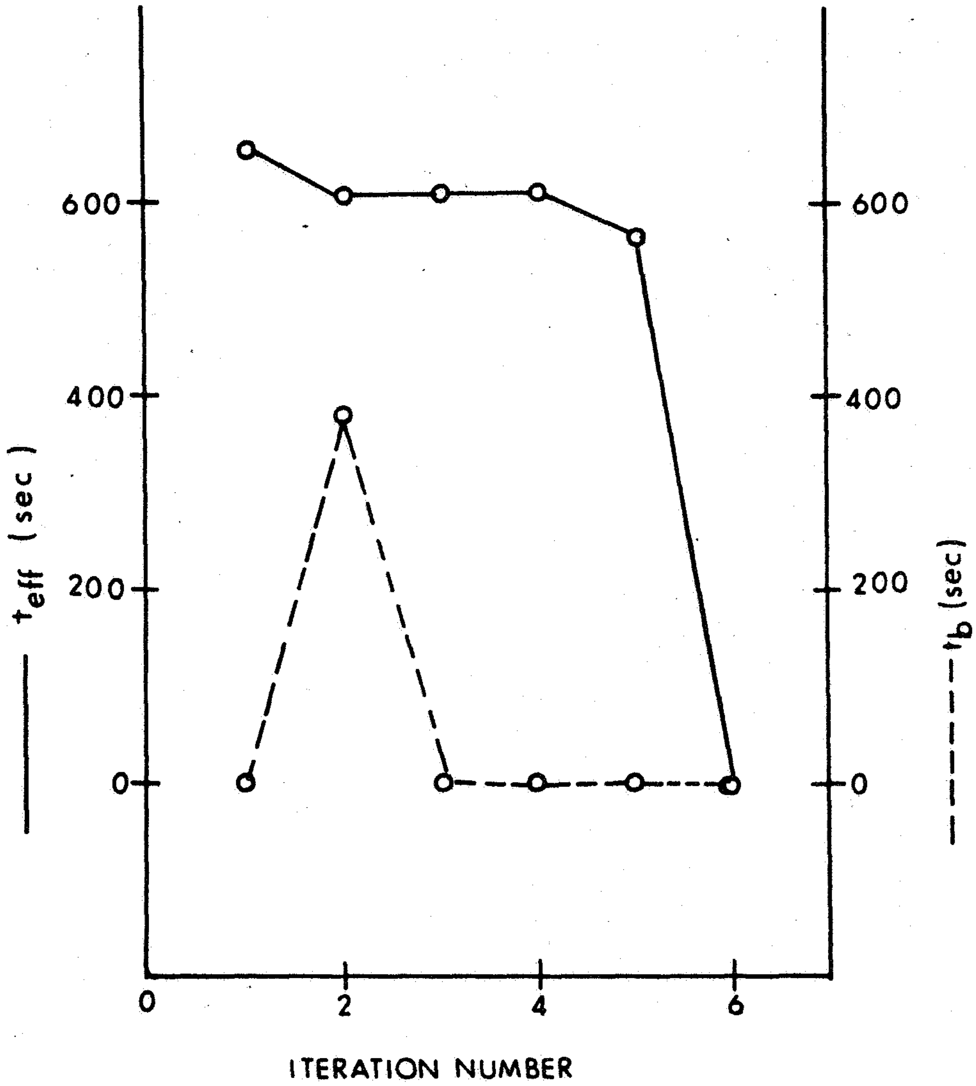
and with the lowering of the desired terminal velocity upper bound from 1000 fps to 900 fps. The results are described in Figs. 6.2.4a - 6.2.4h. The purpose of this case was to lessen the importance of the acceleration penalty such that the desired terminal constraints assumed a larger relative weighting than previously. From a comparison of Figs. 6.2.3a and 6.2.4a, the terminal cost term at the final iteration step has been reduced from 14 to 6, due to improvements in lessening the terminal deviation in velocity and flight path angle. The terminal altitude was satisfactory in both cases. However, these terminal state weightings are still small as compared to the weighting for the acceleration profile, and subsequently, not all of the terminal soft constraints were attained.

The fifth case is the same as the previous, excepting the following:



PALLFS1
300K init. conds.
$W_{g's}$ = 1
W_v = 0.0001
W_γ = 0.01
W_h = 0.000001
η = 2.5
ACCRUED COST $\int L$
TERMINAL COST F
TOTAL COST J
800
600
400
200
0
150
100
50
ITERATION NUMBER
2
4
6

Figure 6. 2. 4a



600
400
200
0
t_{eff} (sec)
600
400
200
0
t_b (sec)
0
2
4
6
ITERATION NUMBER

Figure 6. 2. 4b

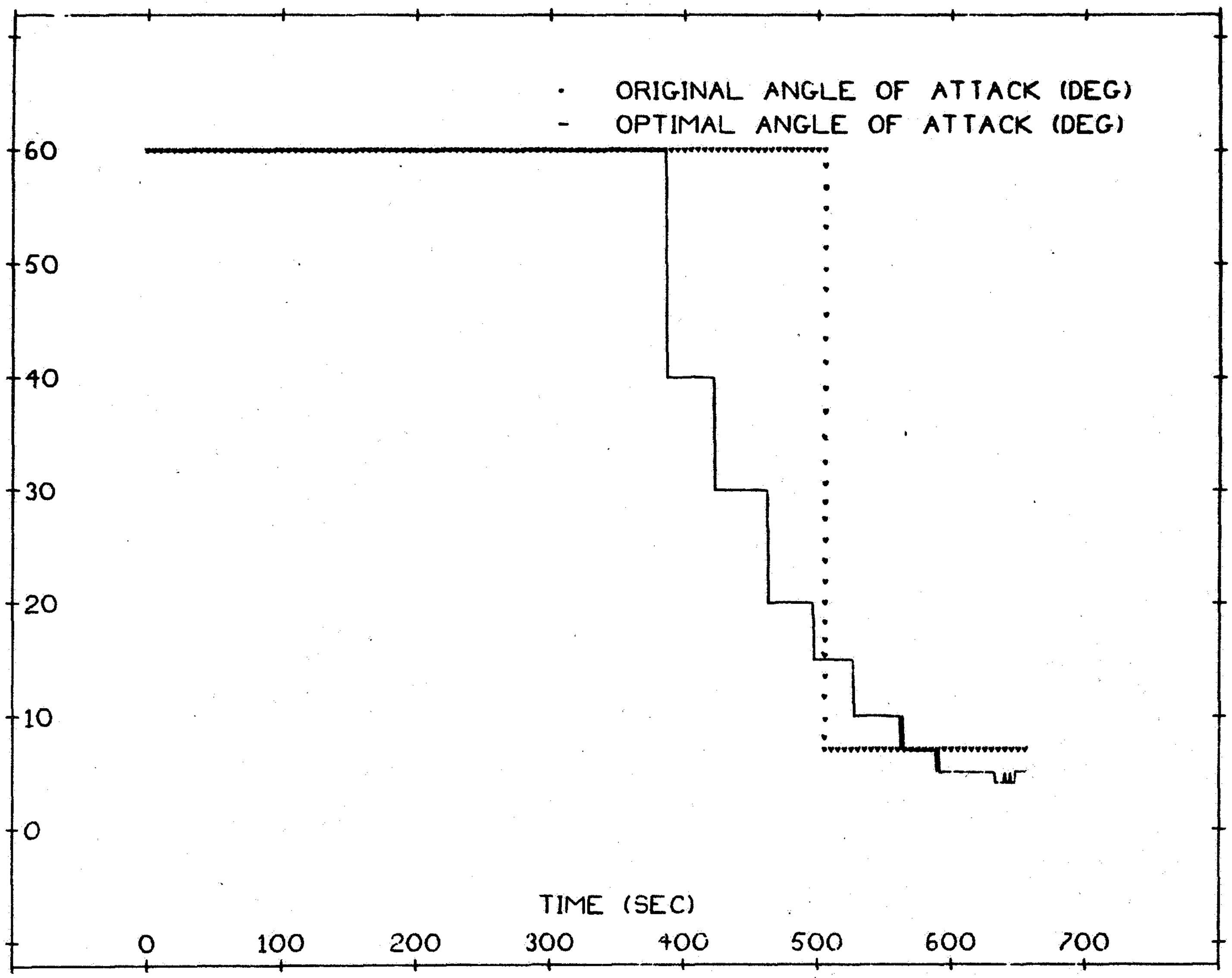
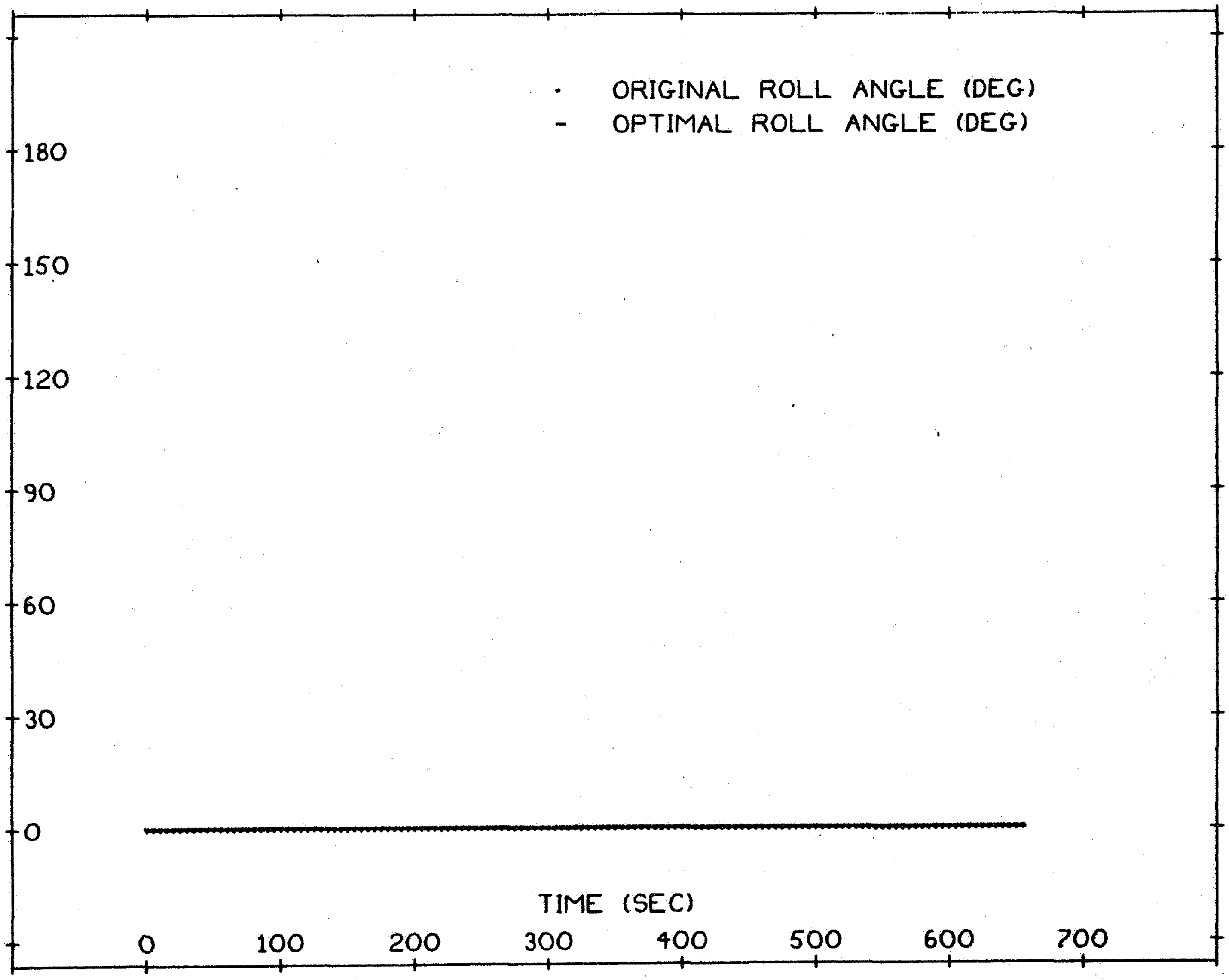

· ORIGINAL ANGLE OF ATTACK (DEG)
- OPTIMAL ANGLE OF ATTACK (DEG)
60
50
40
30
20
10
0
TIME (SEC)
0
100
200
300
400
500
600
700

Figure 6.2.4c



ORIGINAL ROLL ANGLE (DEG)
OPTIMAL ROLL ANGLE (DEG)
180
150
120
90
60
30
0
TIME (SEC)
0
100
200
300
400
500
600
700

Figure 6.2.4d

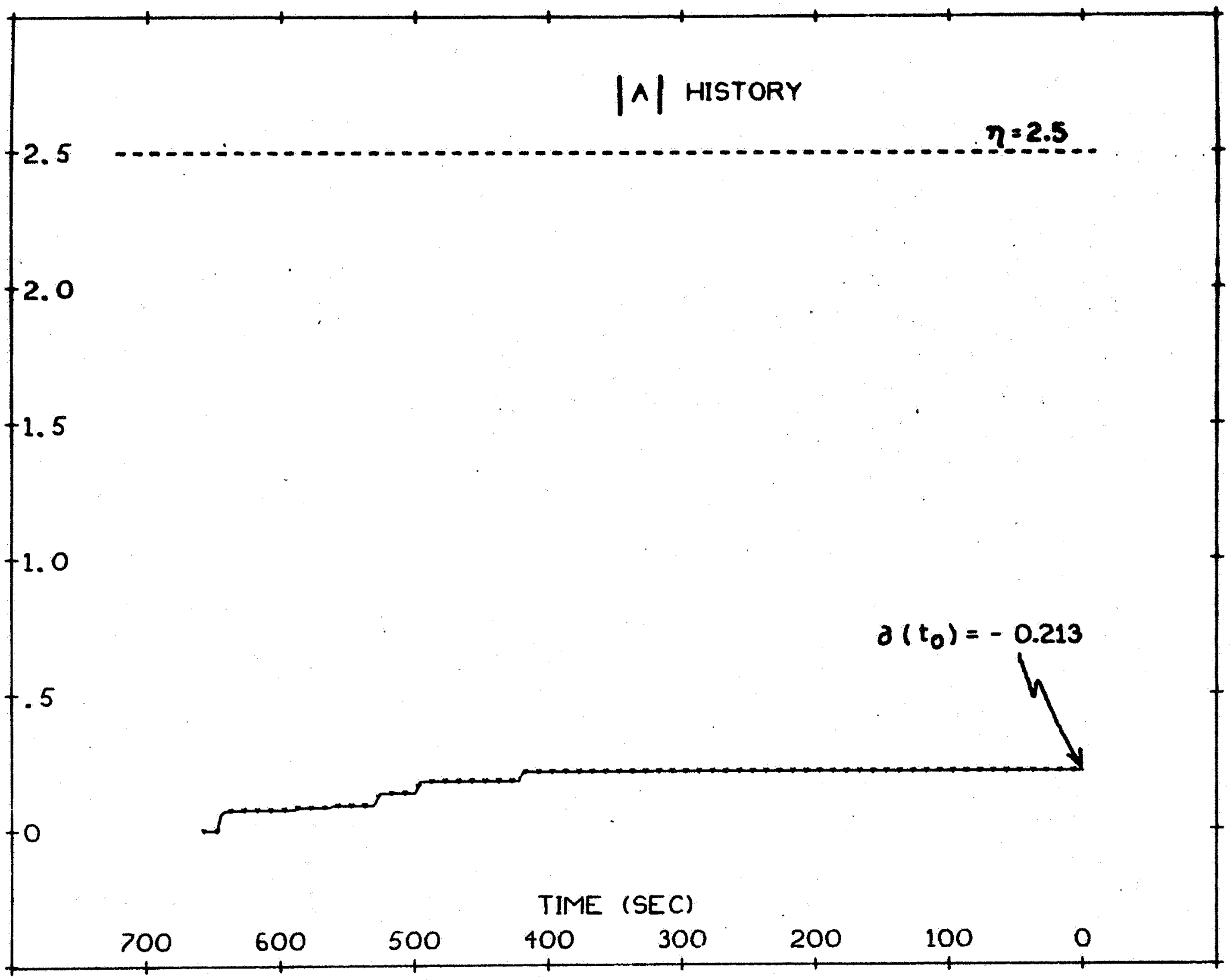
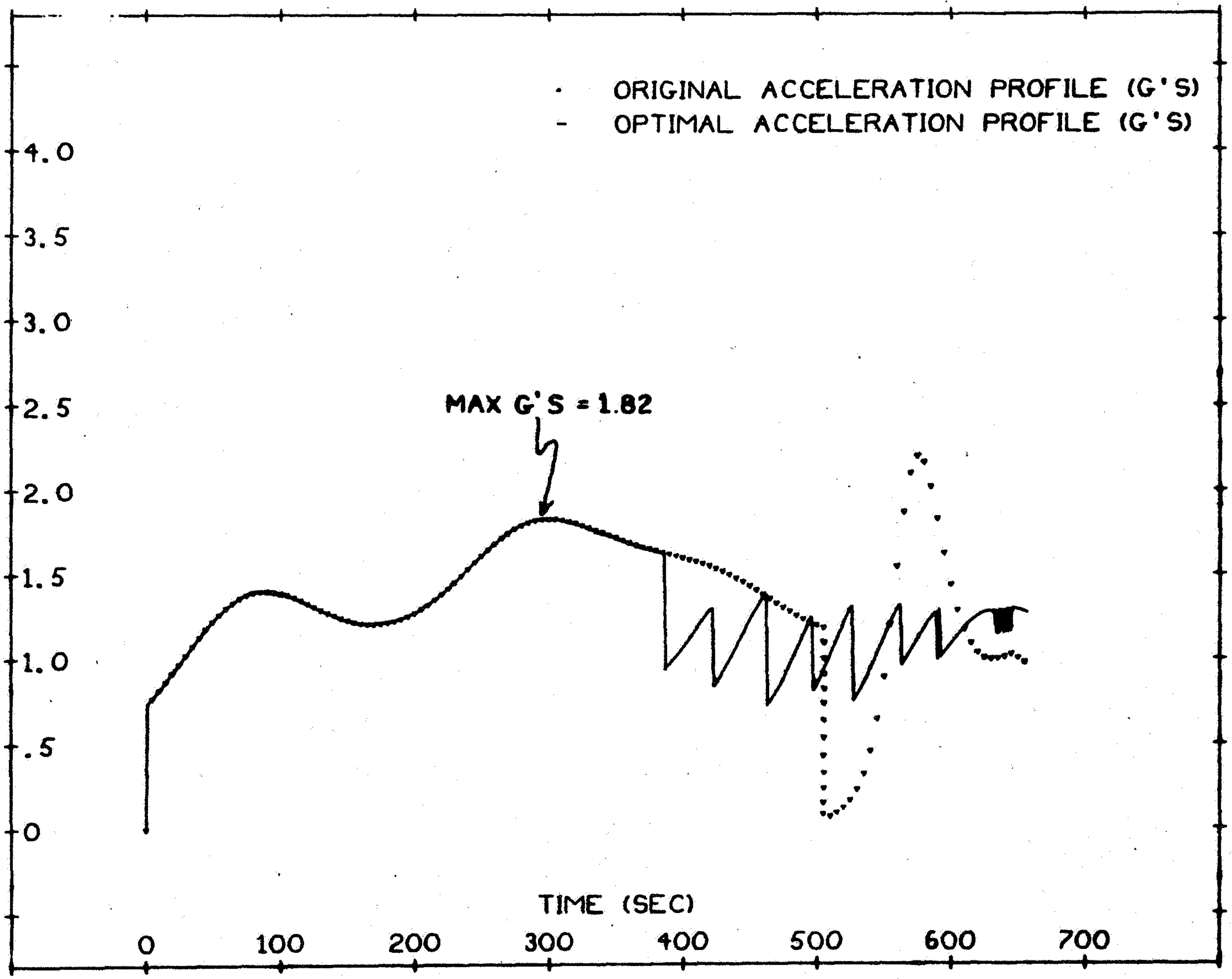
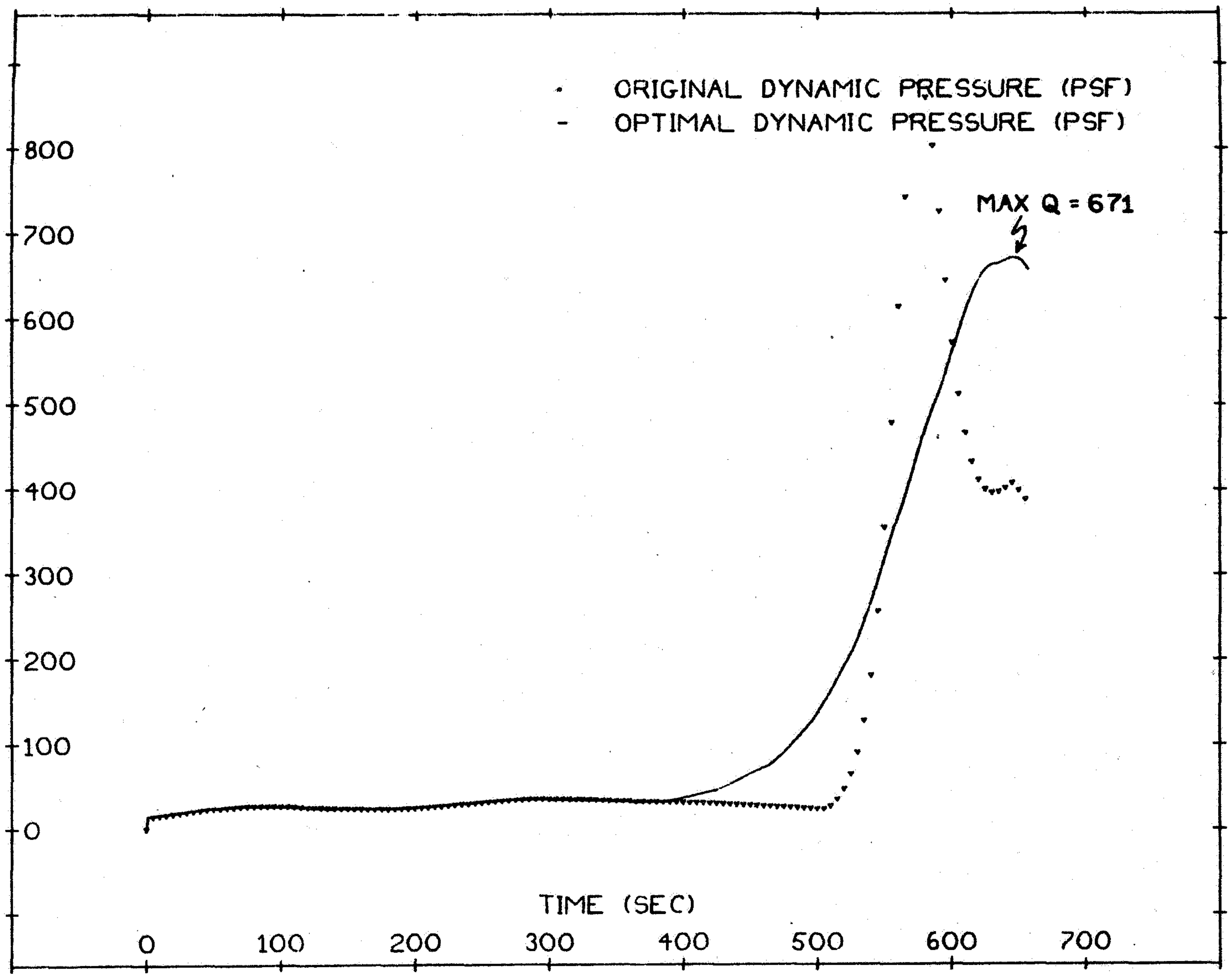

|A| HISTORY
η=2.5
∂(t0) = - 0.213
2.5
2.0
1.5
1.0
.5
0
TIME (SEC)
700
600
500
400
300
200
100
0

Figure 6.2.4e



· ORIGINAL ACCELERATION PROFILE (G'S)
- OPTIMAL ACCELERATION PROFILE (G'S)
MAX G'S = 1.82
4.0
3.5
3.0
2.5
2.0
1.5
1.0
.5
0
TIME (SEC)
0
100
200
300
400
500
600
700

Figure 6.2.4f



ORIGINAL DYNAMIC PRESSURE (PSF)
OPTIMAL DYNAMIC PRESSURE (PSF)
MAX Q = 671
800
700
600
500
400
300
200
100
0
TIME (SEC)
0
100
200
300
400
500
600
700

Figure 6.2.4 g

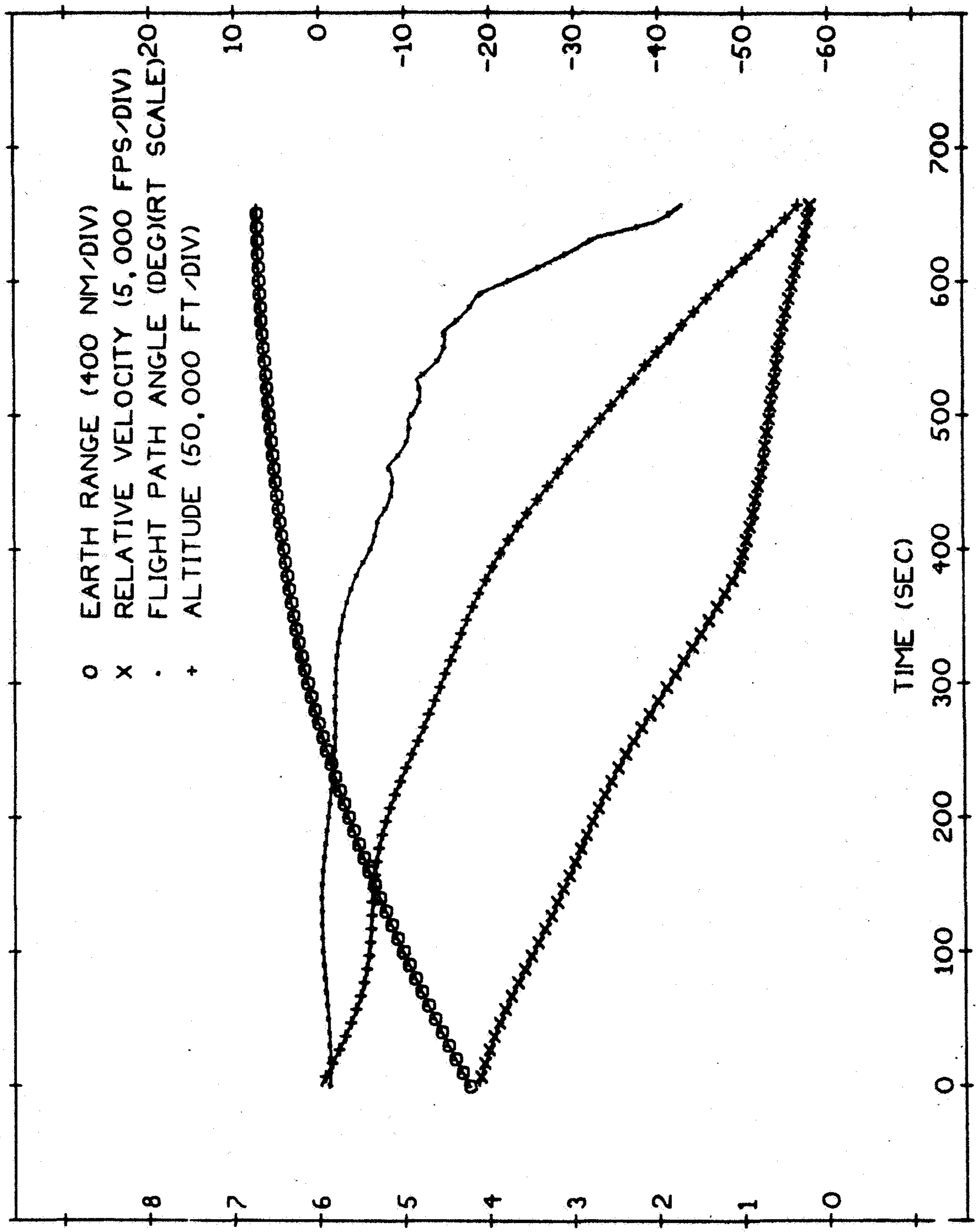
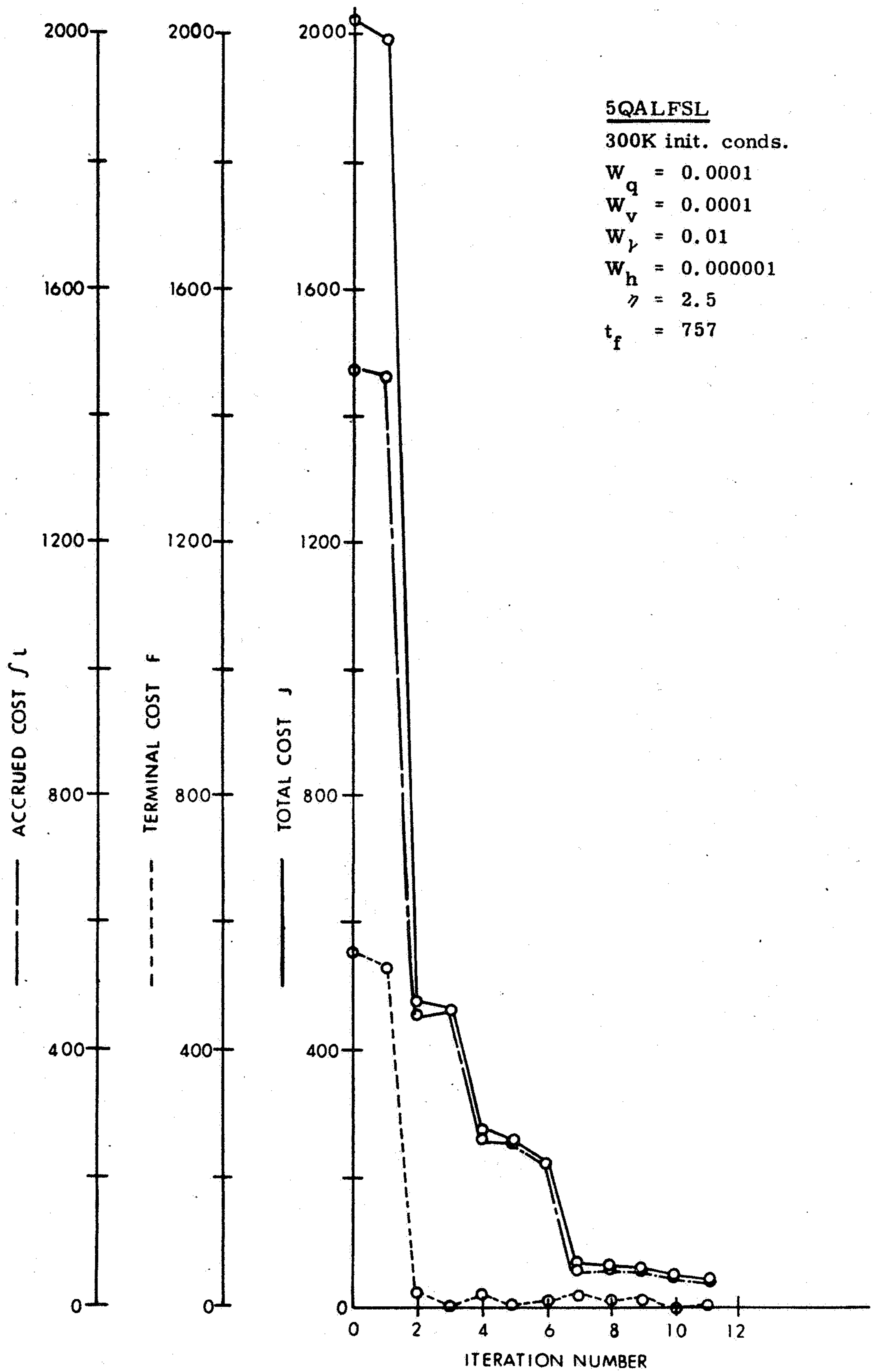

o EARTH RANGE (400 NM/DIV)
x RELATIVE VELOCITY (5,000 FPS/DIV)
· FLIGHT PATH ANGLE (DEG)(RT SCALE)
+ ALTITUDE (50,000 FT/DIV)
TIME (SEC)
0
100
200
300
400
500
600
700
0
1
2
3
4
5
6
7
8
20
10
0
-10
-20
-30
-40
-50
-60

Figure 6.2.4h



2000
1600
1200
800
400
0
ACCRUED COST ∫L
TERMINAL COST F
TOTAL COST J
0
2
4
6
8
10
12
ITERATION NUMBER
5QALFSL
300K init. conds.
W_q = 0.0001
W_v = 0.0001
W_γ = 0.01
W_h = 0.000001
η = 2.5
t_f = 757

Figure 6. 2. 5 a

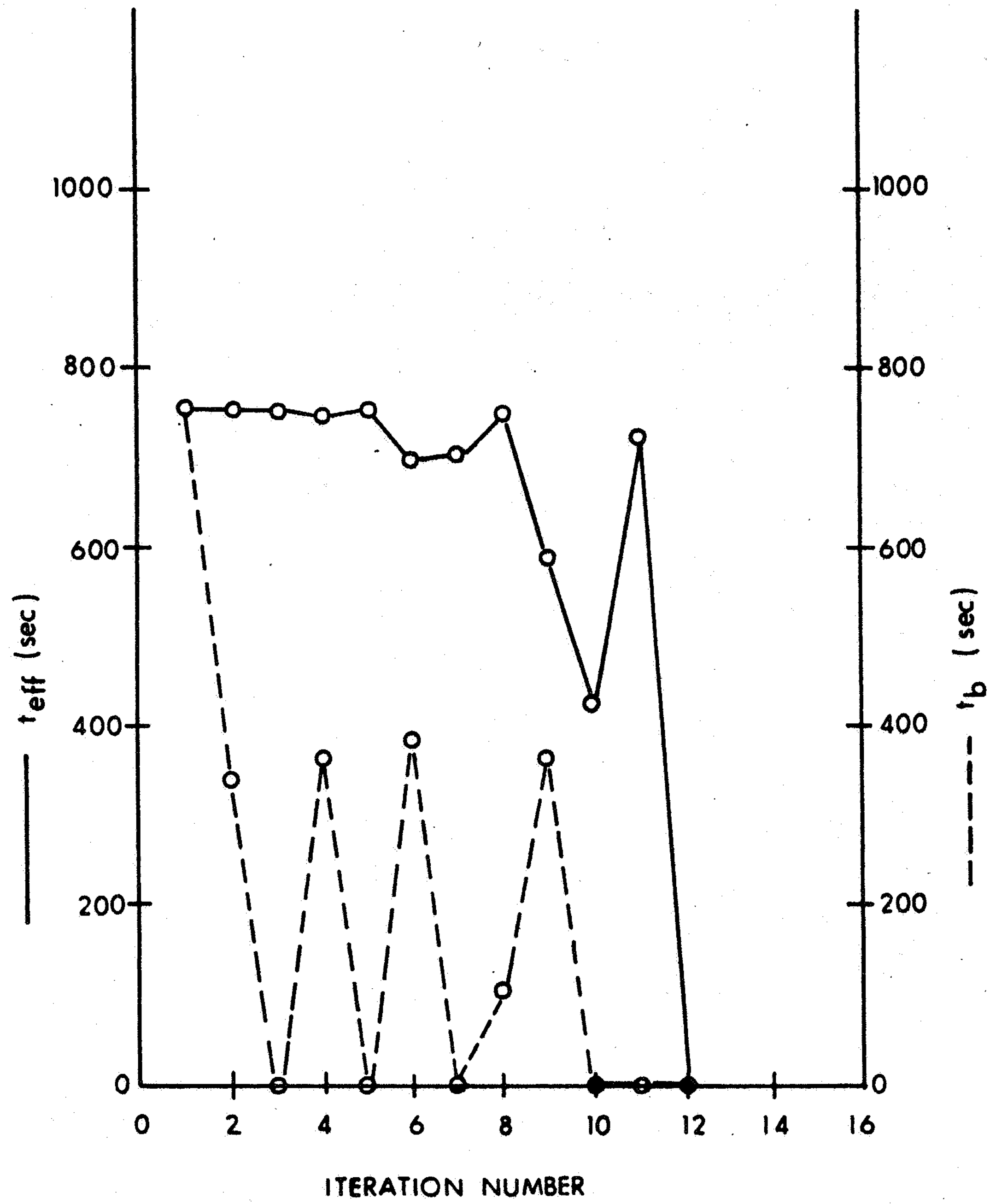
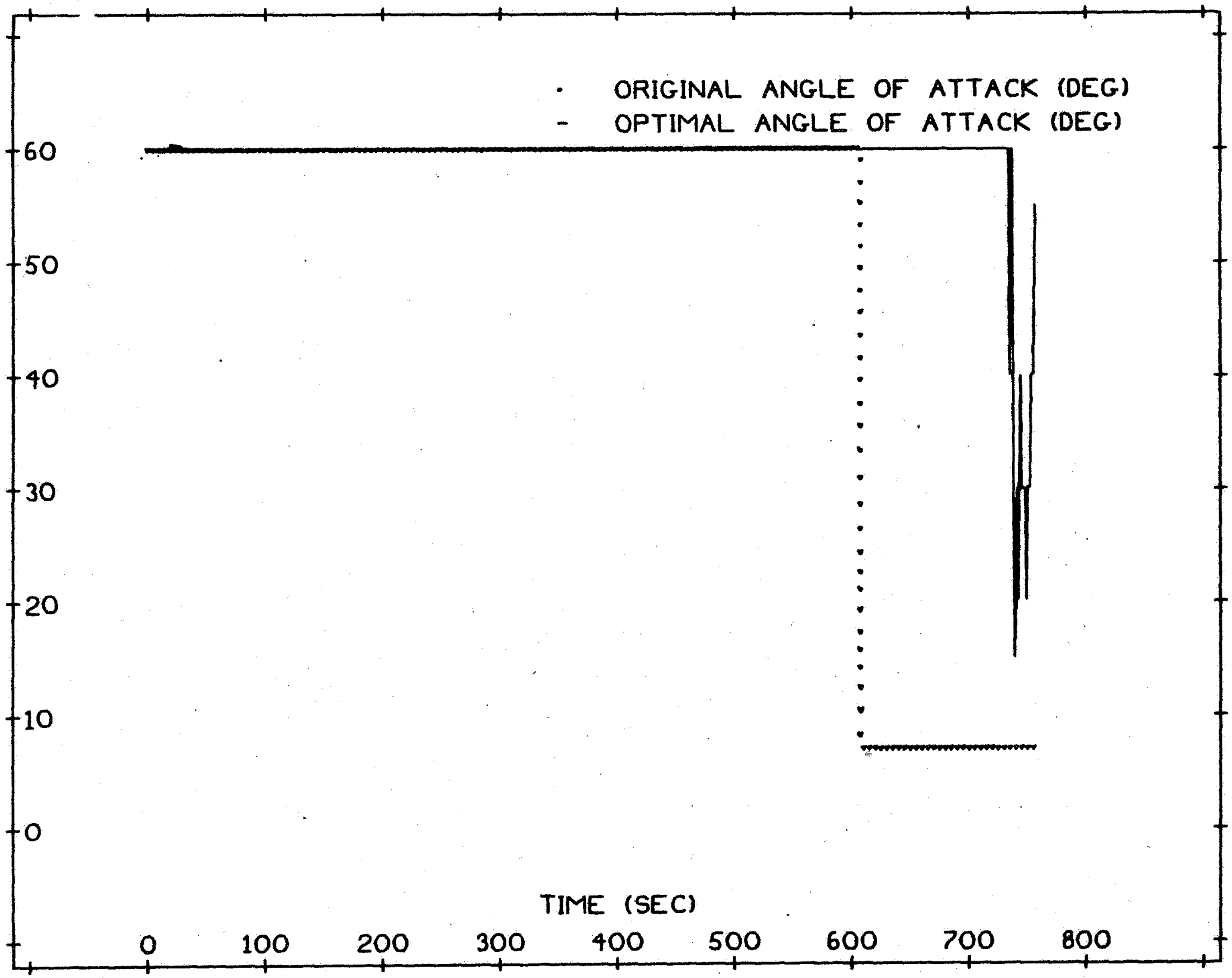
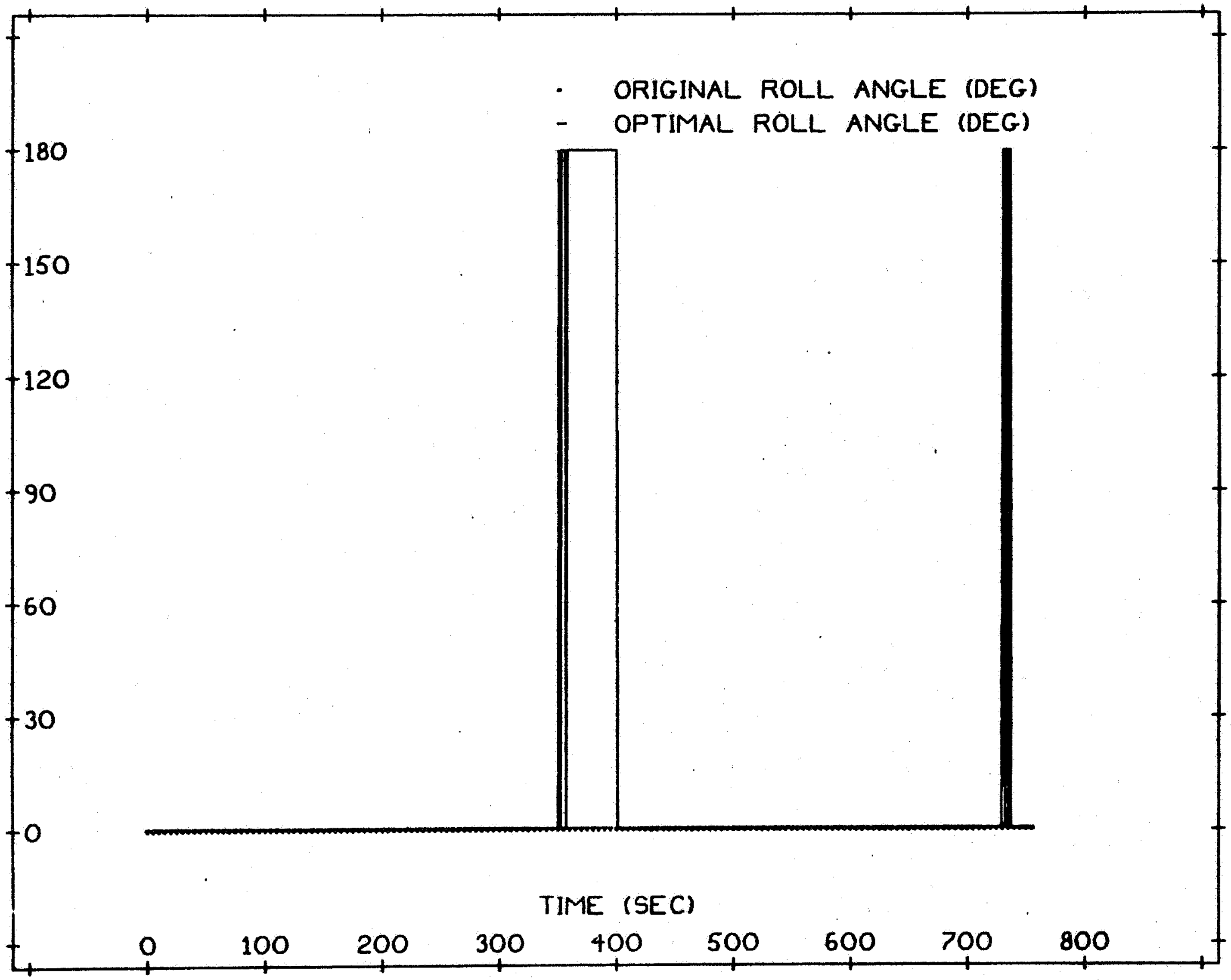

1000
800
600
400
200
0
0
2
4
6
8
10
12
14
16
ITERATION NUMBER
t_{eff} (sec)
t_b (sec)

Figure 6. 2. 5b



ORIGINAL ANGLE OF ATTACK (DEG)
OPTIMAL ANGLE OF ATTACK (DEG)
60
50
40
30
20
10
0
TIME (SEC)
0
100
200
300
400
500
600
700
800

Figure 6.2.5c



ORIGINAL ROLL ANGLE (DEG)
OPTIMAL ROLL ANGLE (DEG)
180
150
120
90
60
30
0
TIME (SEC)
0
100
200
300
400
500
600
700
800

Figure 6.2.5d

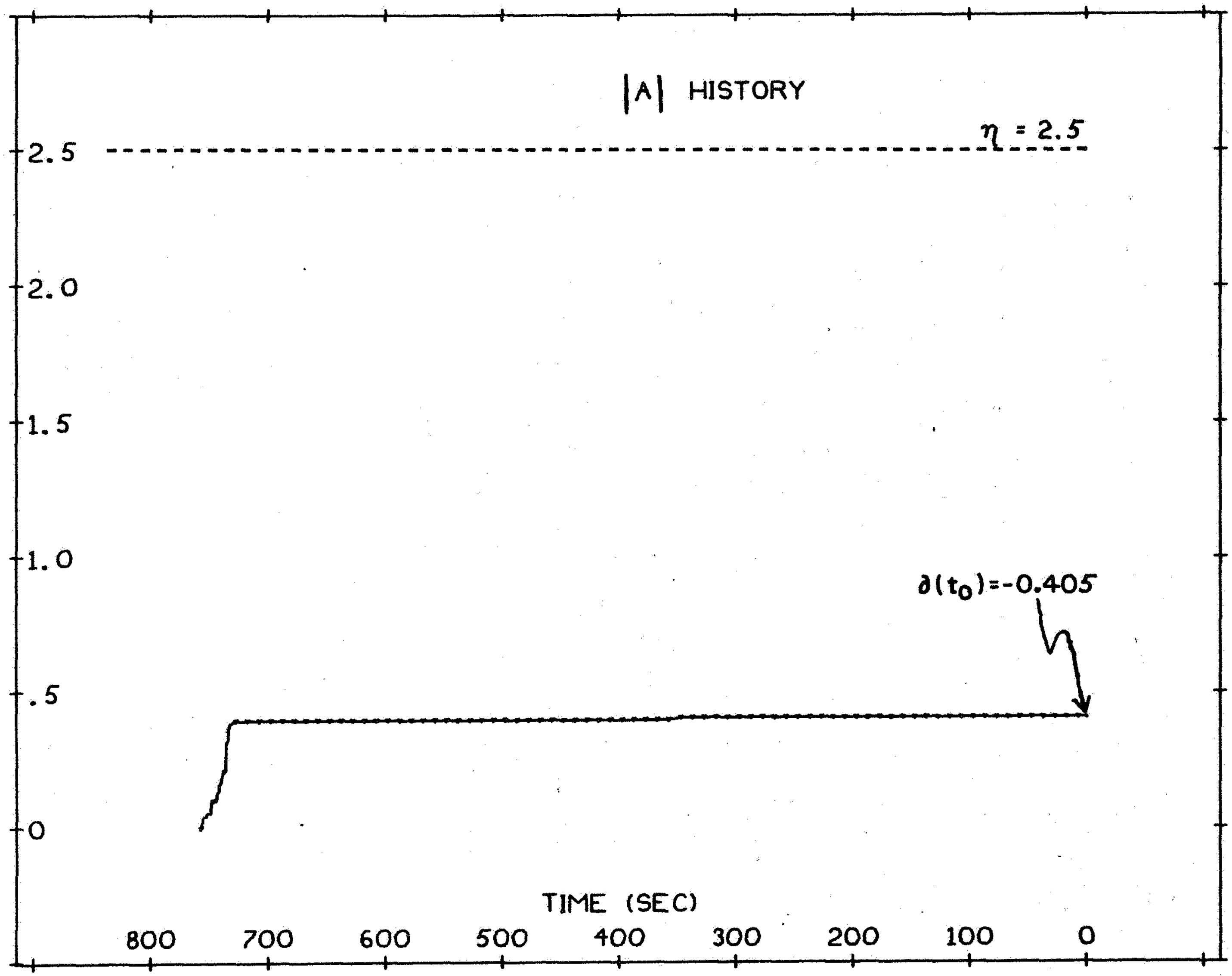

|A| HISTORY
η = 2.5
$\partial(t_0)$=-0.405
2.5
2.0
1.5
1.0
.5
0
TIME (SEC)
800
700
600
500
400
300
200
100
0

Figure 6.2.5e

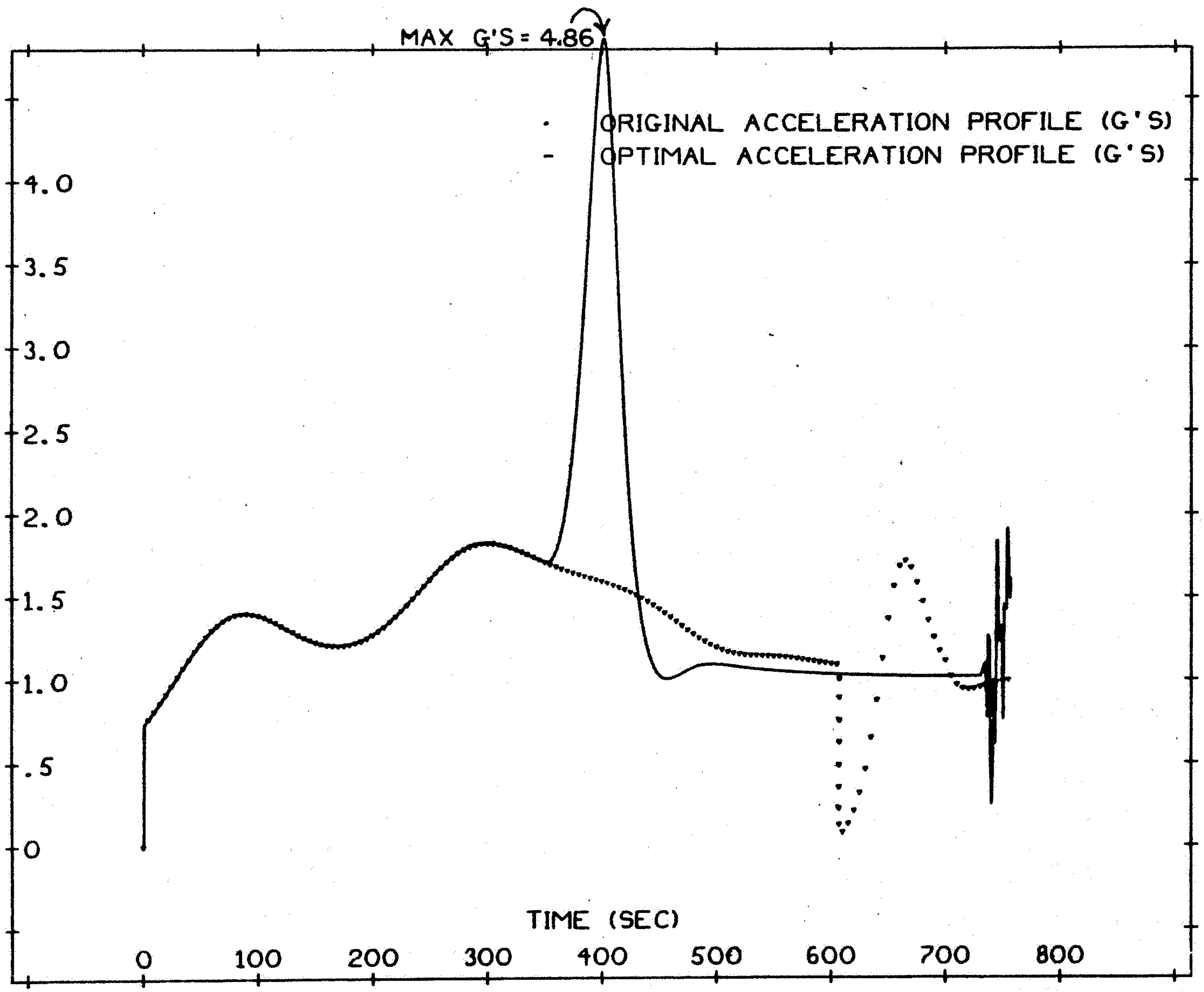
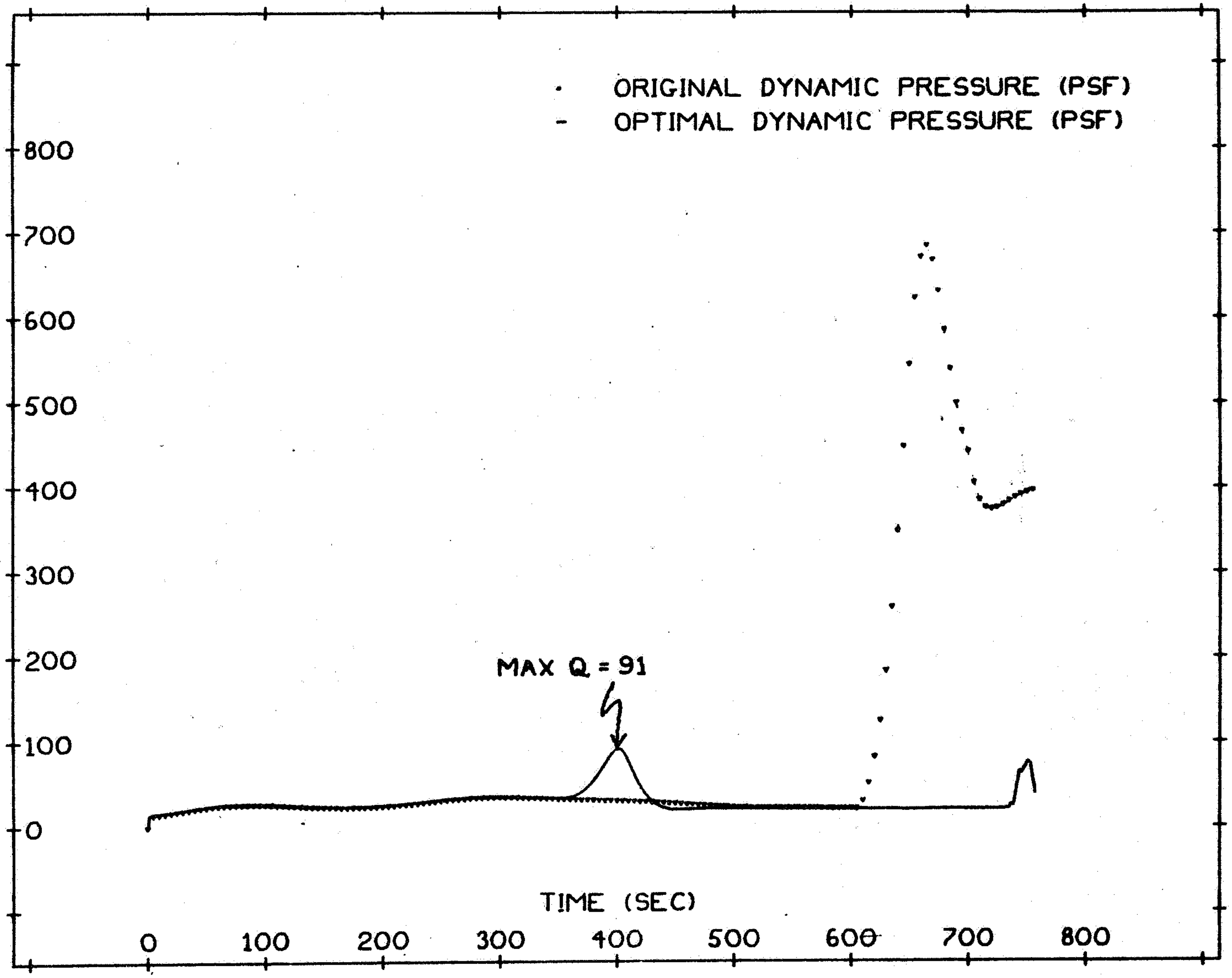
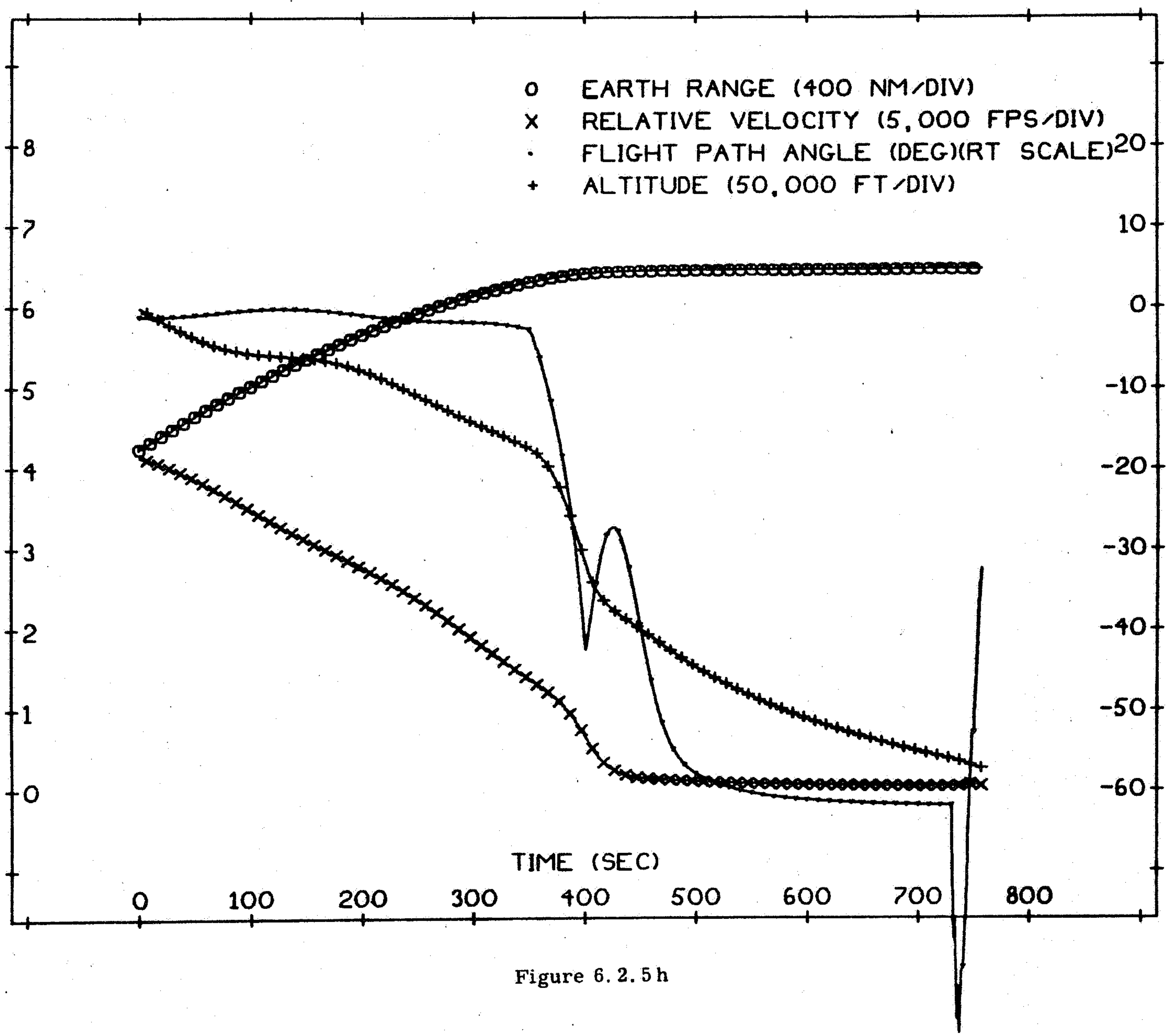

MAX G'S = 4.86
· ORIGINAL ACCELERATION PROFILE (G'S)
- OPTIMAL ACCELERATION PROFILE (G'S)
4.0
3.5
3.0
2.5
2.0
1.5
1.0
.5
0
TIME (SEC)
0
100
200
300
400
500
600
700
800

Figure 6.2.5f



ORIGINAL DYNAMIC PRESSURE (PSF)
OPTIMAL DYNAMIC PRESSURE (PSF)
MAX Q = 91
TIME (SEC)
0
100
200
300
400
500
600
700
800

Figure 6.2.5 g



o EARTH RANGE (400 NM/DIV)
x RELATIVE VELOCITY (5,000 FPS/DIV)
· FLIGHT PATH ANGLE (DEG)(RT SCALE)
+ ALTITUDE (50,000 FT/DIV)
8
7
6
5
4
3
2
1
0
20
10
0
-10
-20
-30
-40
-50
-60
TIME (SEC)
0
100
200
300
400
500
600
700
800

Figure 6.2.5h

$$W_{g's} = 0 \qquad (6.2.27)$$

$$W_q = .0001 \qquad (6.2.28)$$

Also, the terminal time was extended from 657 sec to 757 sec. This is equivalent to a 1200 sec. trajectory from an initial altitude of 400,000 ft. Figures 6.2.5a - 6.2.5h depict these results. From Fig. 6.2.5a, the terminal cost is now reduced to 2, with the only terminal violation being in the flight path angle, but now with a still smaller deviation from the desired values than in the previous two cases. Figures 6.2.5f and 6.2.5g show the acceleration and dynamic pressure profiles, respectively, where the former is essentially ignored while the latter is penalized. In this case, the optimal dynamic pressure history has been cost wise reduced to where it has a maximum peak of only 91 lbs/ft^2. This is to be compared with the dynamic pressure contours of Figs. 6.2.3g and 6.2.4g. Figure 6.2.5h reveals the optimal state variable histories.

The sixth case differs from the previous in the following:

$$W_{g's} = .25 \qquad (6.2.29)$$

$$W_q = 0 \qquad (6.2.30)$$

$$\eta = 2.0 \qquad (6.2.31)$$

$$t_f = 657 \qquad (6.2.32)$$

It is essentially equivalent to the third and fourth cases, but stressing the accrued penalty term on the acceleration profile still less. In addition, η has been reduced to 2.0. Figures 6.2.6a - 6.2.6h display the results. In Figure 6.2.6a, the terminal cost is shown to be reduced to 0.5, as compared with 14 and 6 in Figs. 6.2.3a and 6.2.4a, respectively. Barring the significance of a slight 6 fps terminal velocity violation, the only significant deviation is in the terminal flight path angle, now smaller

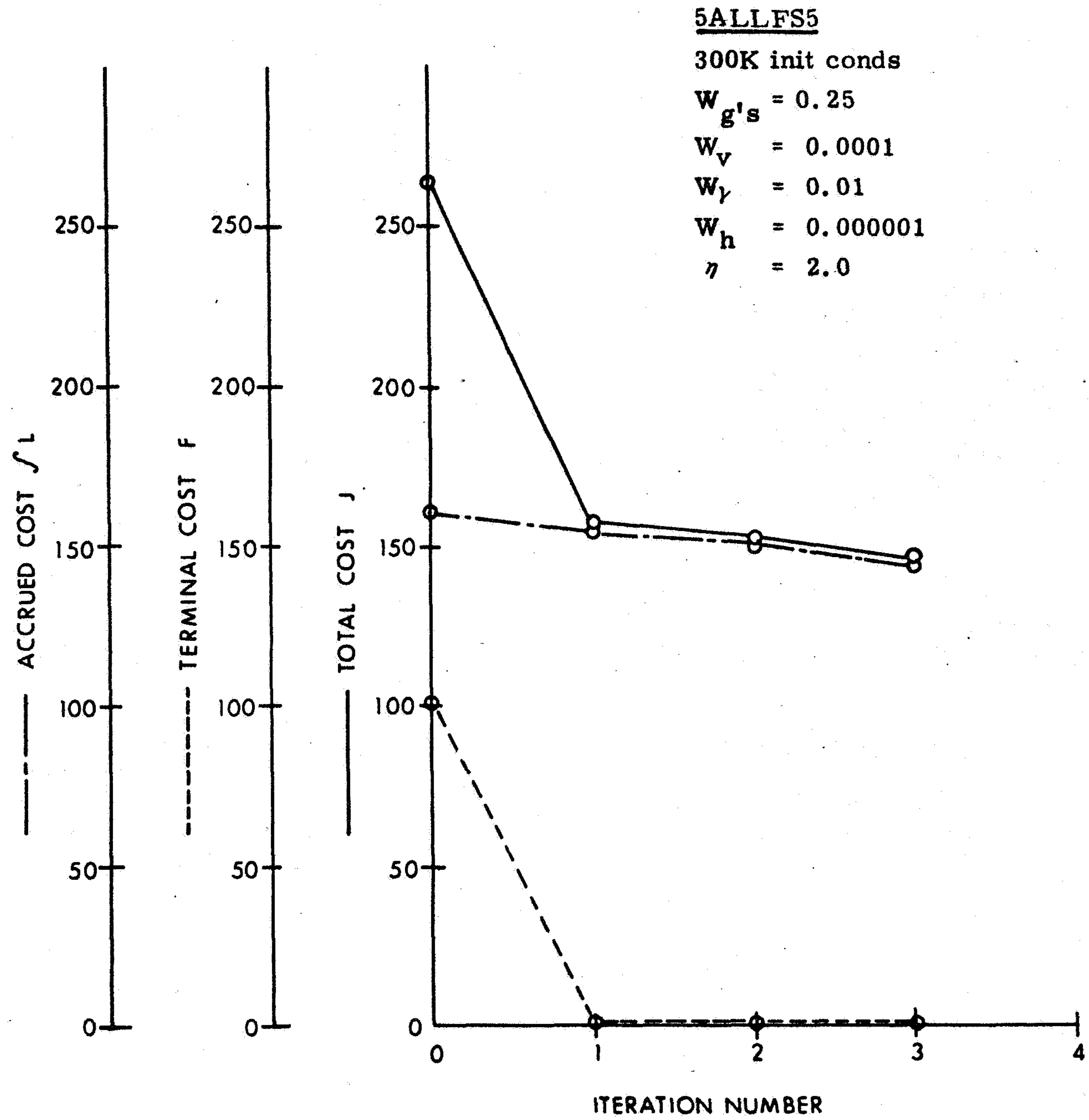

5ALLFS5
300K init conds
$W_{g's}$ = 0.25
W_v = 0.0001
W_γ = 0.01
W_h = 0.000001
η = 2.0
ACCRUED COST $\int L$
TERMINAL COST F
TOTAL COST J
ITERATION NUMBER
0
50
100
150
200
250
1
2
3
4

Figure 6.2.6a

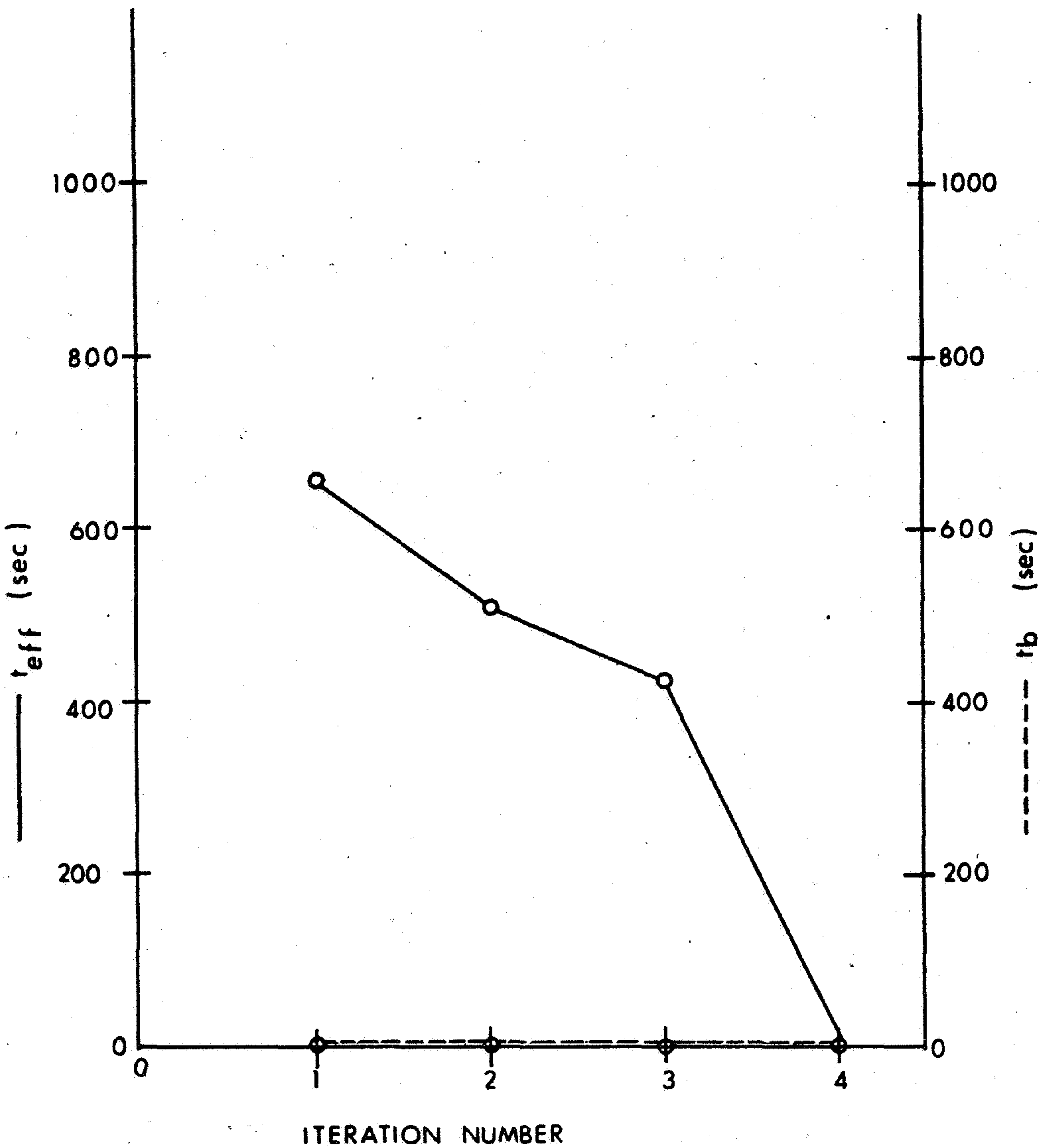
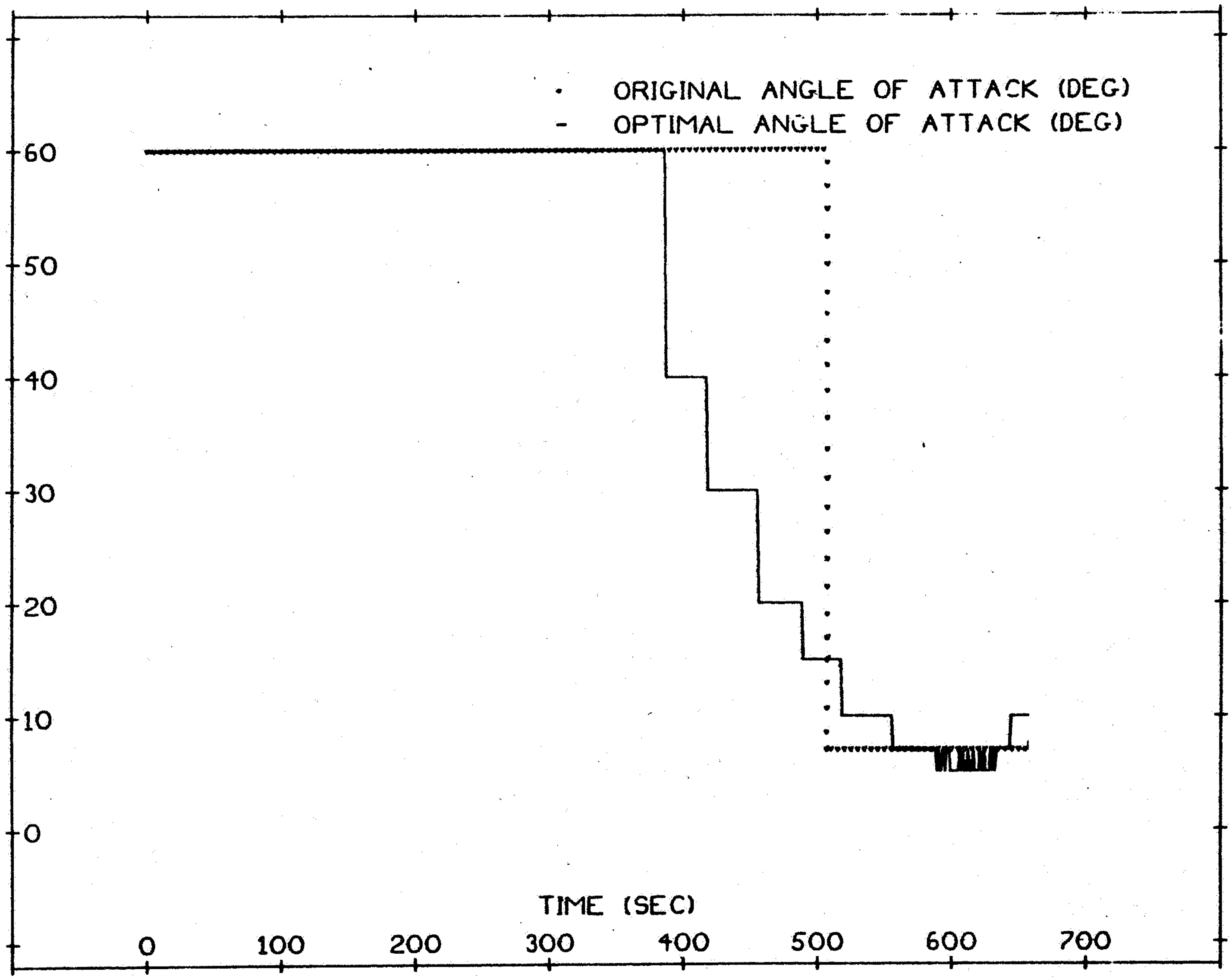
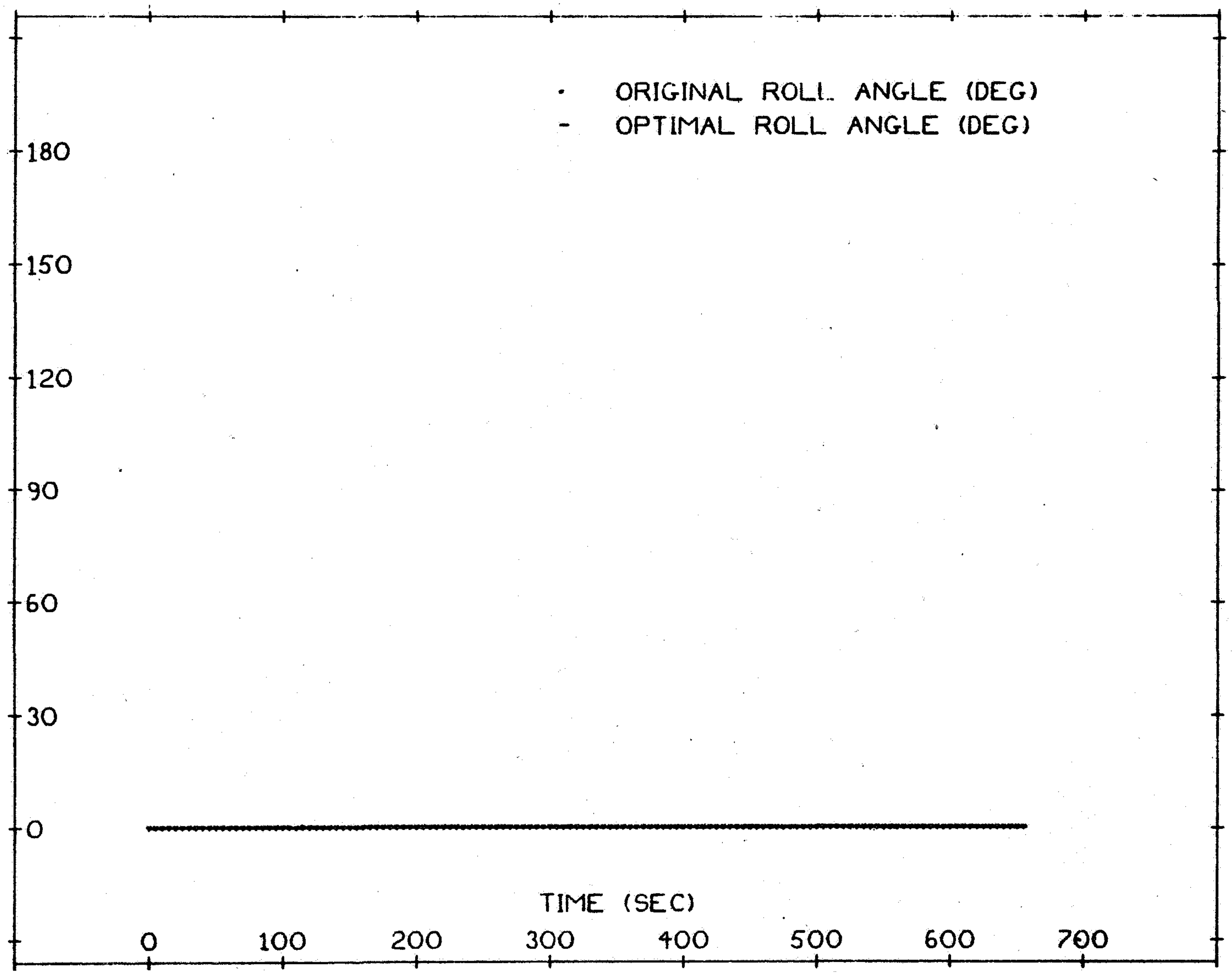


Figure 6.2.6b



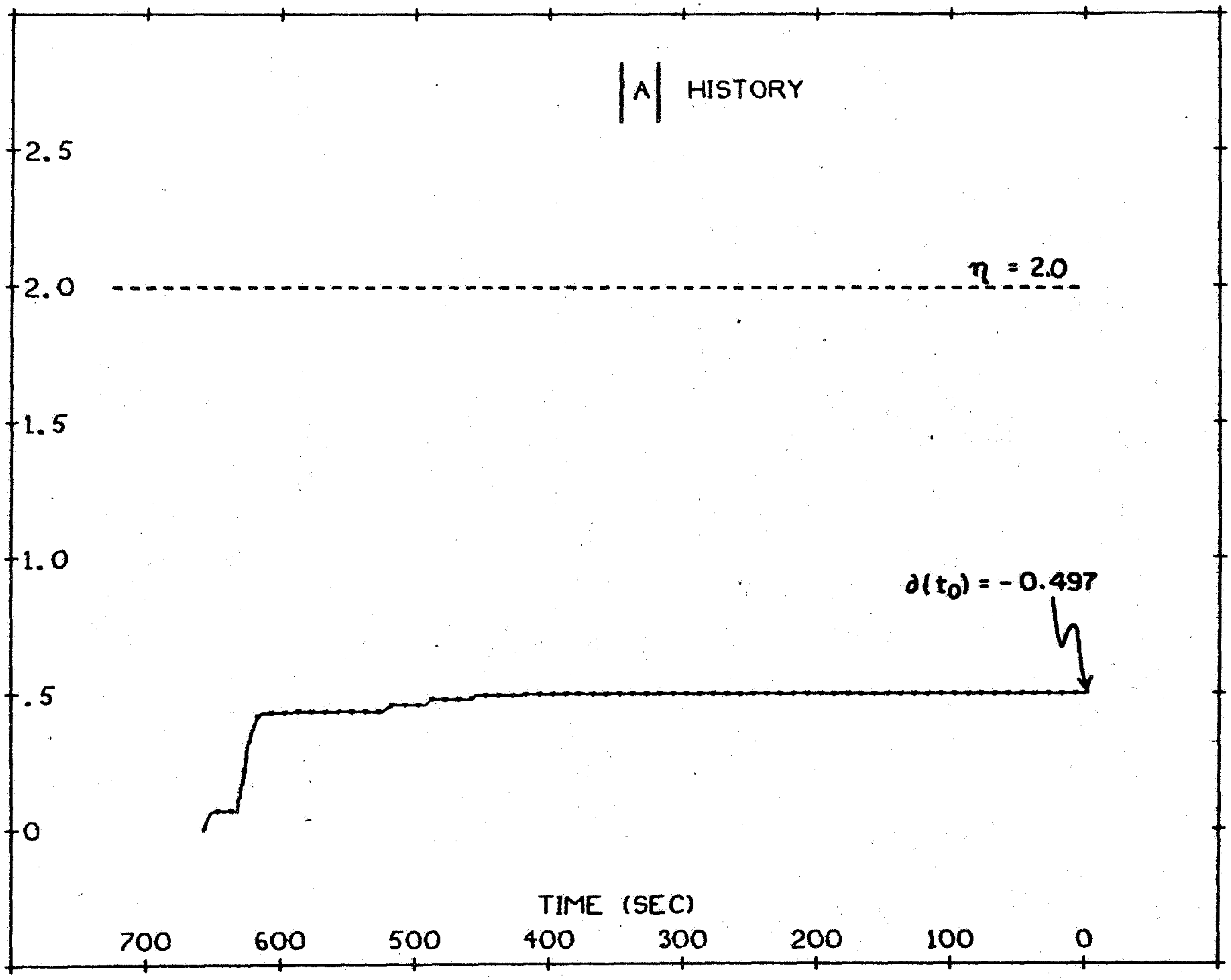
· ORIGINAL ANGLE OF ATTACK (DEG)
- OPTIMAL ANGLE OF ATTACK (DEG)
60
50
40
30
20
10
0
TIME (SEC)
0
100
200
300
400
500
600
700

Figure 6.2.6c



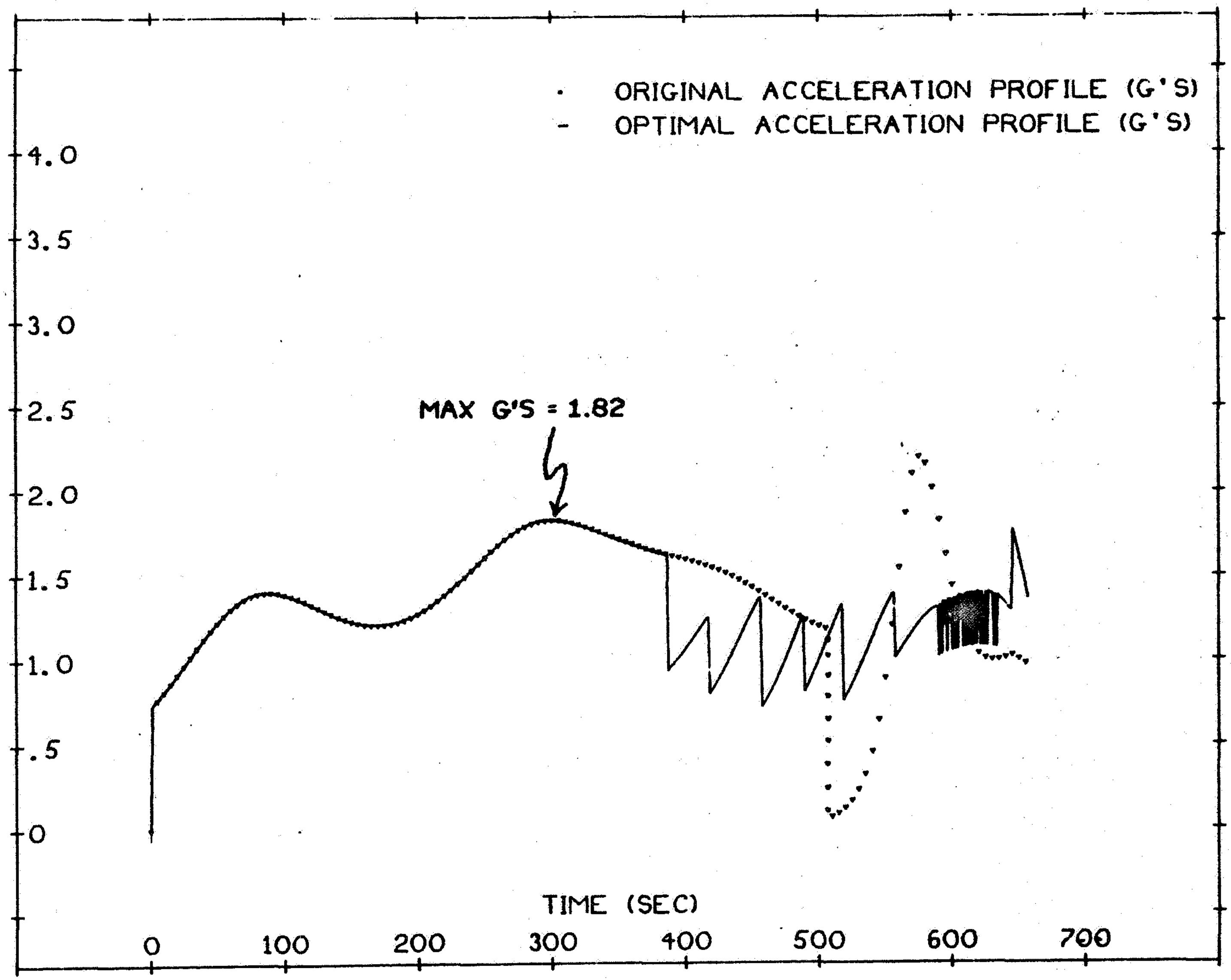
· ORIGINAL ROLL ANGLE (DEG)
- OPTIMAL ROLL ANGLE (DEG)
180
150
120
90
60
30
0
TIME (SEC)
0
100
200
300
400
500
600
700

Figure 6.2.6d



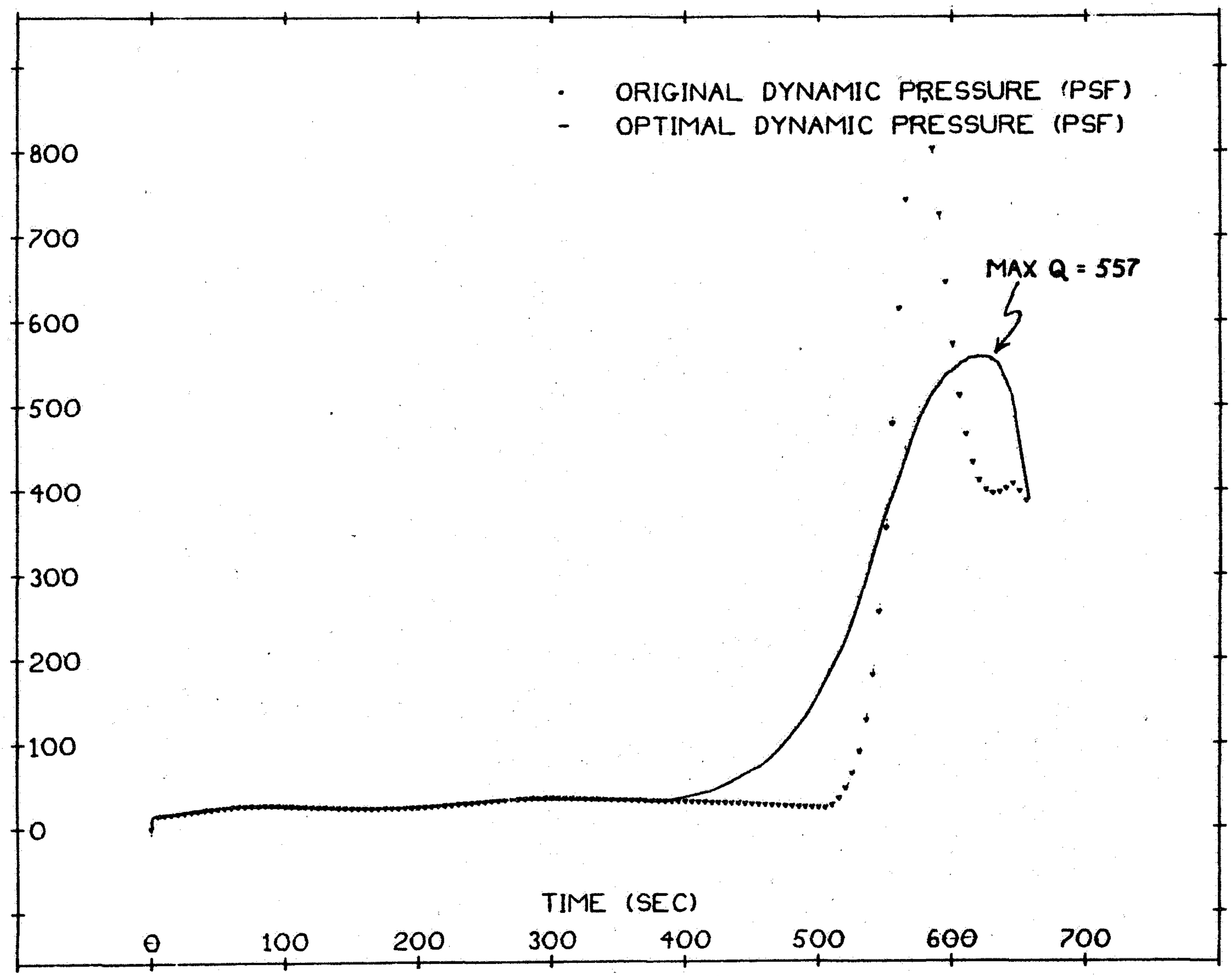
|A| HISTORY
2.5
2.0
1.5
1.0
.5
0
η = 2.0
$\partial(t_0) = -0.497$
TIME (SEC)
700
600
500
400
300
200
100
0

Figure 6.2.6e



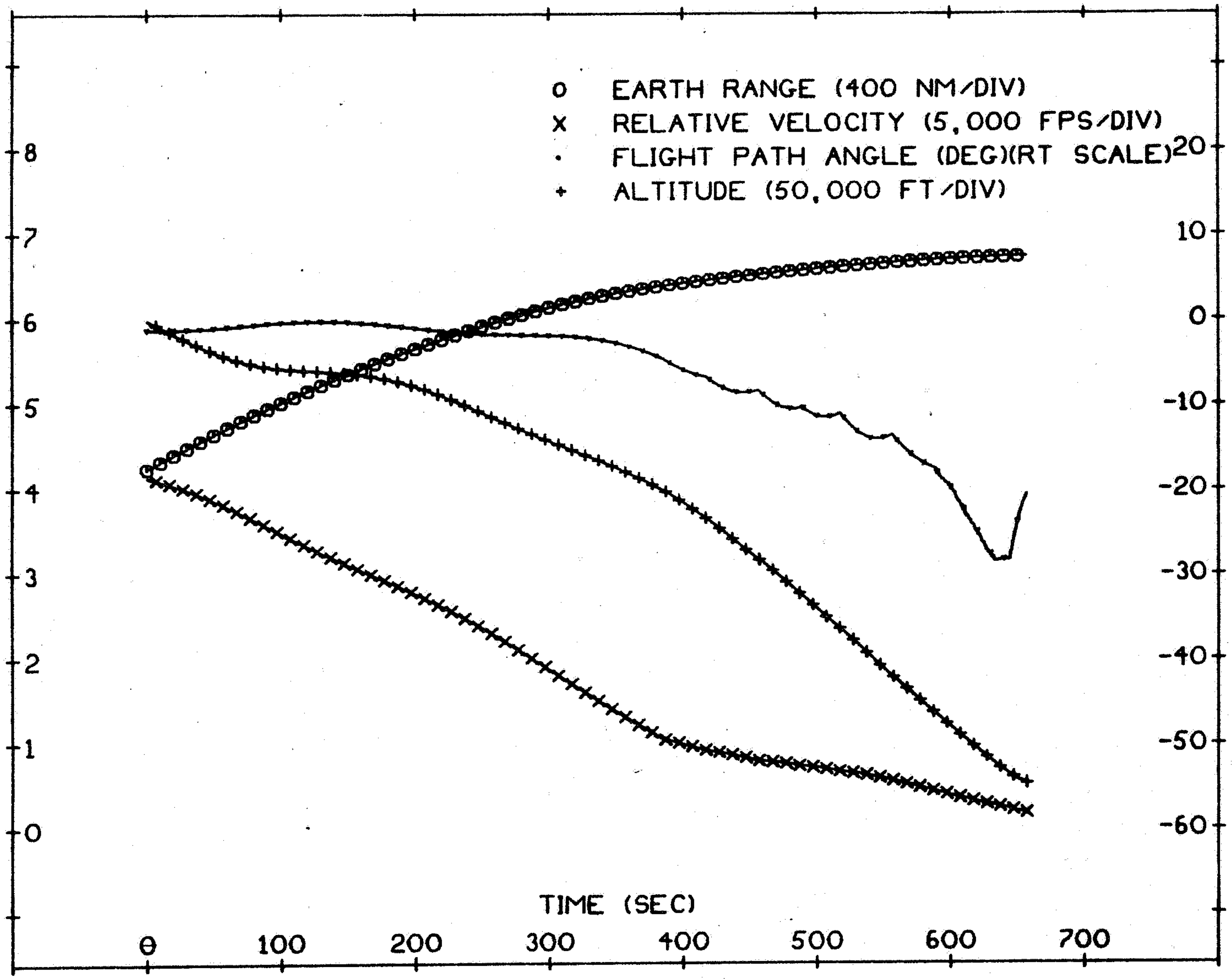
· ORIGINAL ACCELERATION PROFILE (G'S)
- OPTIMAL ACCELERATION PROFILE (G'S)
MAX G'S = 1.82
4.0
3.5
3.0
2.5
2.0
1.5
1.0
.5
0
TIME (SEC)
0
100
200
300
400
500
600
700

Figure 6.2.6f



ORIGINAL DYNAMIC PRESSURE (PSF)
OPTIMAL DYNAMIC PRESSURE (PSF)
MAX Q = 557
800
700
600
500
400
300
200
100
0
TIME (SEC)
0
100
200
300
400
500
600
700

Figure 6.2.6 g



o EARTH RANGE (400 NM/DIV)
x RELATIVE VELOCITY (5,000 FPS/DIV)
· FLIGHT PATH ANGLE (DEG)(RT SCALE)
+ ALTITUDE (50,000 FT/DIV)
8
7
6
5
4
3
2
1
0
20
10
0
-10
-20
-30
-40
-50
-60
TIME (SEC)
0
100
200
300
400
500
600
700

Figure 6.2.6h

than the terminal flight path angle deviations of the previous three cases. Thus, with an even smaller weighting on the accrued cost term than in the fourth case, the relative importance of the terminal performance is further accentuated. In addition, the shortened time interval has also served to eliminate instabilities in the matrix Riccati equations as evidenced by Fig. 6.2.6b.

A case penalizing both the acceleration profile and the dynamic pressure history is given in Figs. 6.2.7a - 6.2.7h, with the following changes from the previous situation:

$$W_{g's} = 1 \tag{6.2.33}$$

$$W_q = .0001 \tag{6.2.34}$$

$$\eta = 2.5 \tag{6.2.35}$$

With the same terminal state tolerance weightings as in the previous four cases, it is clear that terminal violations are less important than reductions in acceleration and dynamic pressure than in either of the fourth, fifth, or sixth cases where the terminal costs were 6, 2, and 0.5, respectively. Here the final terminal cost is 9 at the seventh iteration as indicated in Fig. 6.2.7a. The shortened time interval did not eliminate the instability in the Riccati equation which appears on the first iteration at t = 313 sec. as evidenced in Fig. 6.2.7b. However, the instability does not appear again in the subsequent iterations. Figures 6.2.7f - 6.2.7g reveal the maximum acceleration to be 1.82 g's and a dynamic pressure peak of 78 lb/ft^2, both quite reasonable and due to the weightings given them in Eqs. (6.2.33) and (6.2.34). Here, terminal state accuracy has been sacrificed in favor of acceptability in both the acceleration and dynamic pressure profiles. The optimal state histories are finally shown in Fig. 6.2.7h.

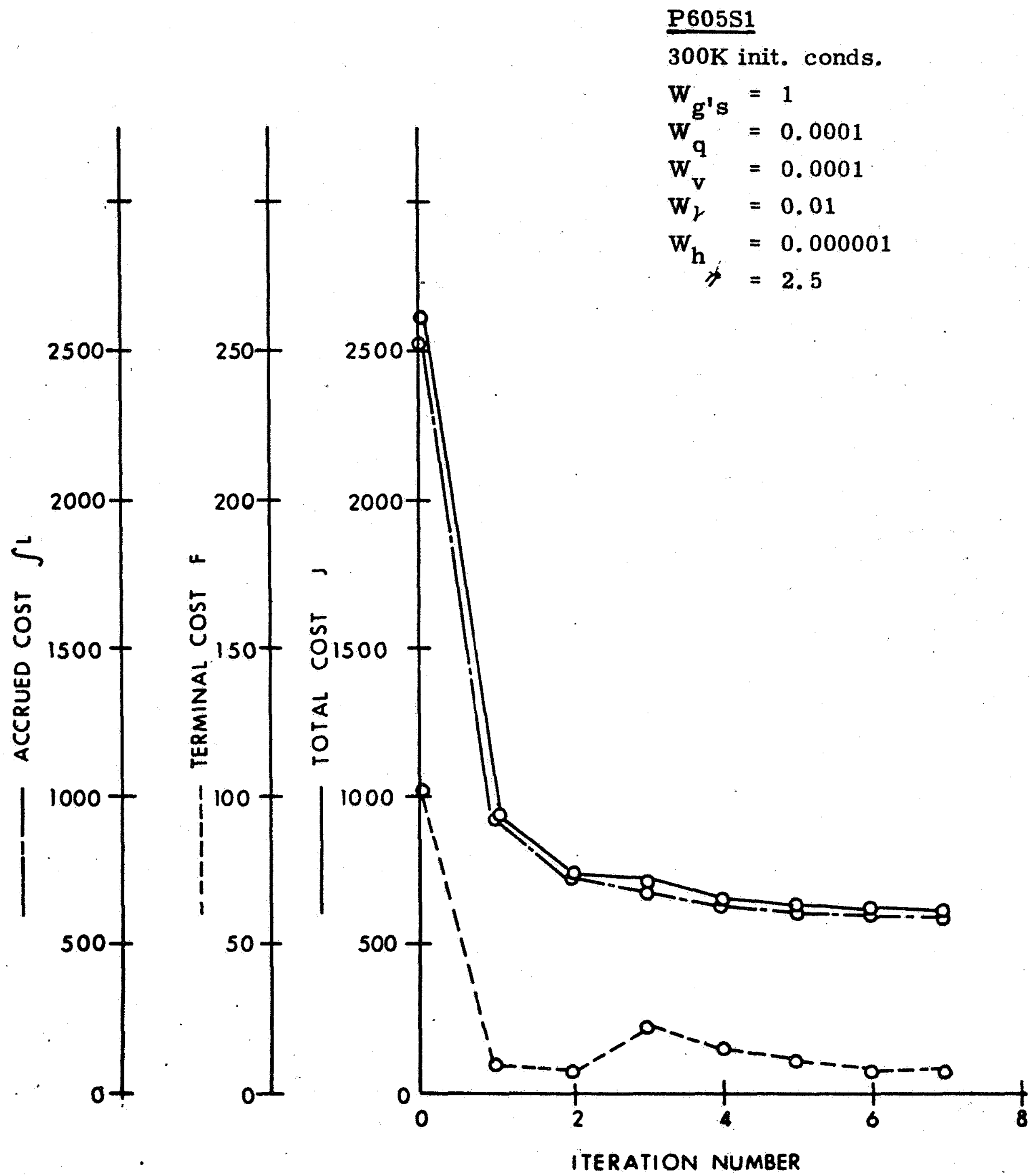

P605S1
300K init. conds.
$W_{g's}$ = 1
W_q = 0.0001
W_v = 0.0001
W_γ = 0.01
W_h = 0.000001
γ = 2.5
ACCRUED COST $\int L$
TERMINAL COST F
TOTAL COST J
2500
2000
1500
1000
500
0
250
200
150
100
50
0
2500
2000
1500
1000
500
0
0
2
4
6
8
ITERATION NUMBER

Figure 6.2.7a

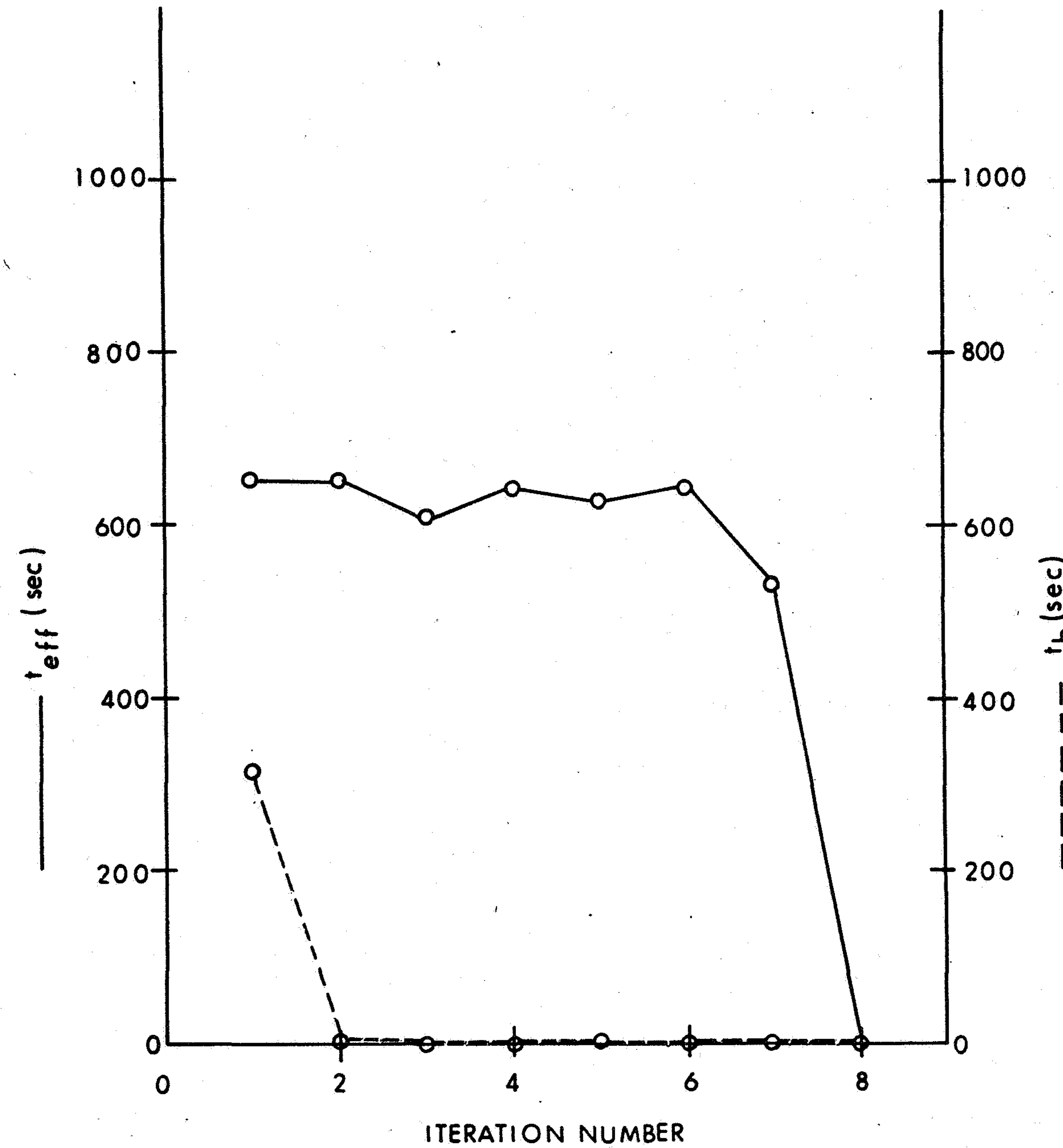
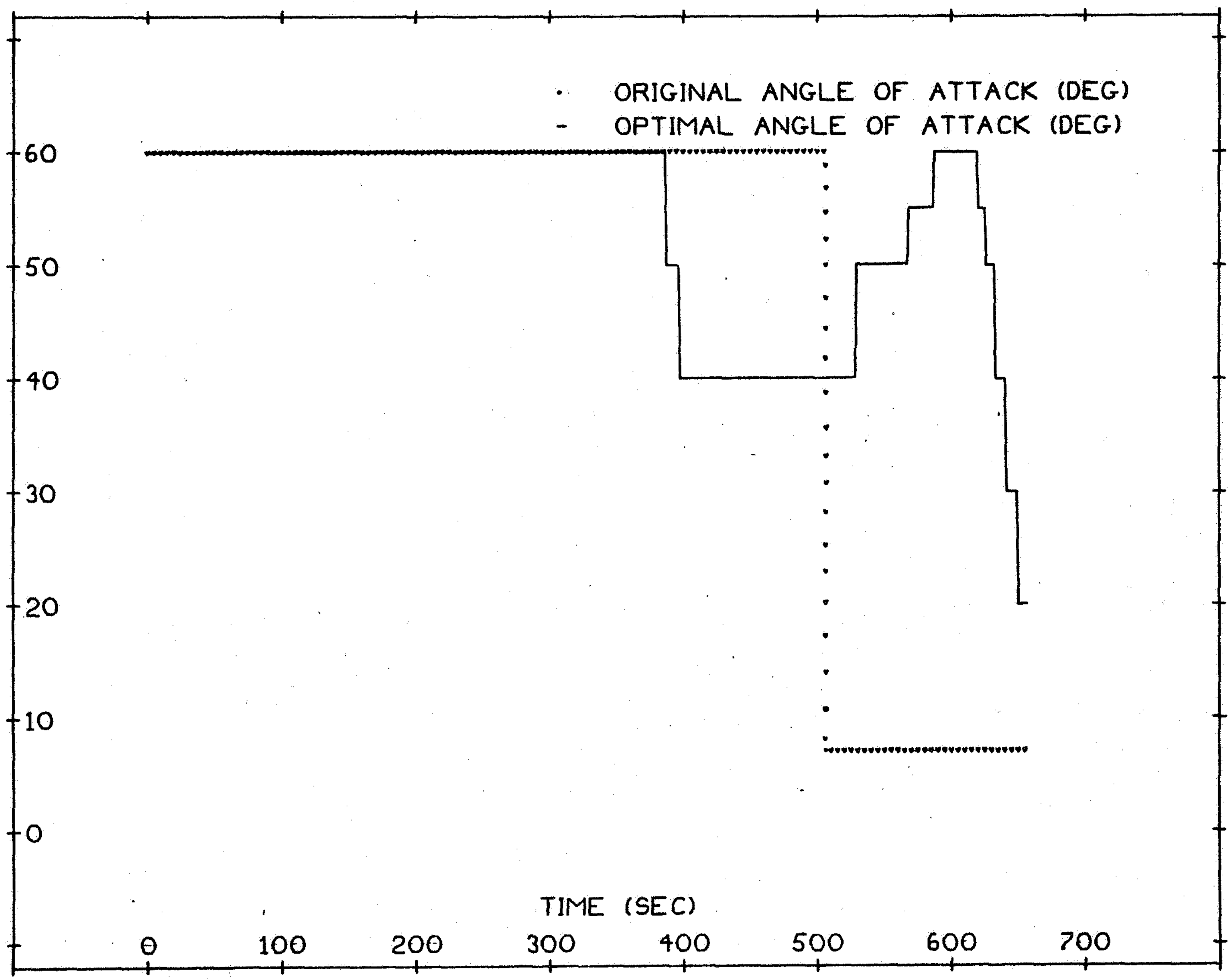
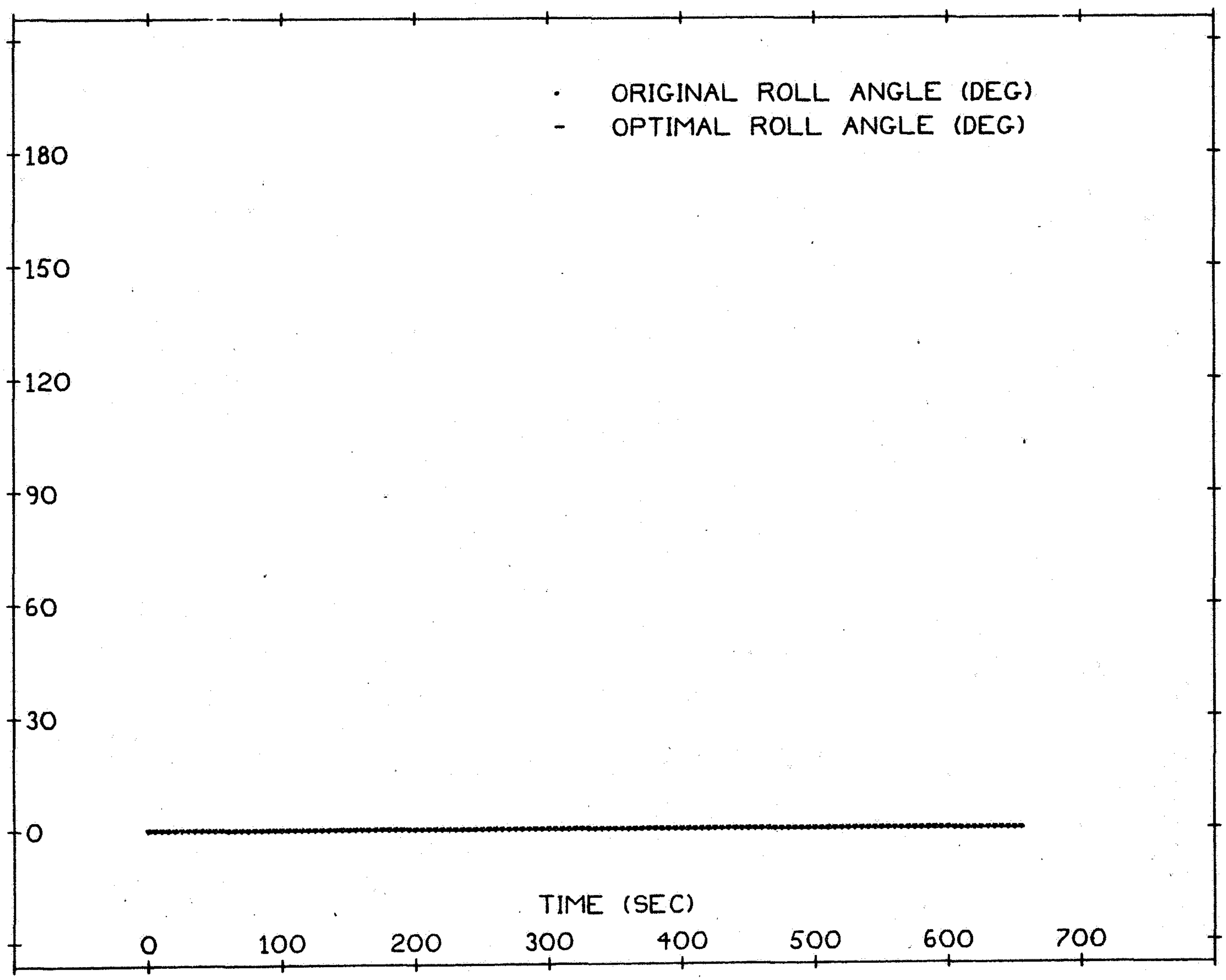


Figure 6.2.7b



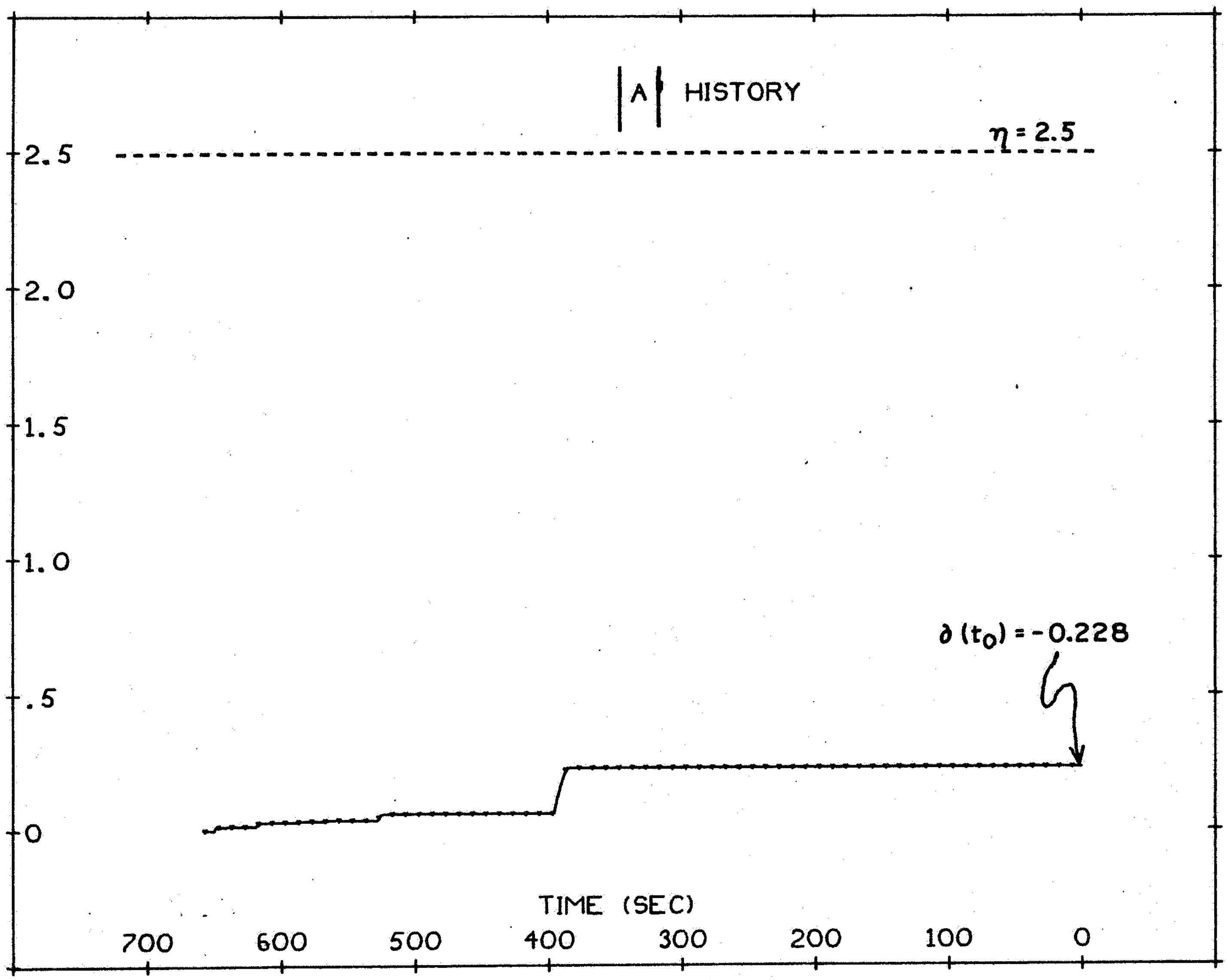
· ORIGINAL ANGLE OF ATTACK (DEG)
- OPTIMAL ANGLE OF ATTACK (DEG)
60
50
40
30
20
10
0
TIME (SEC)
0
100
200
300
400
500
600
700

Figure 6.2.7c



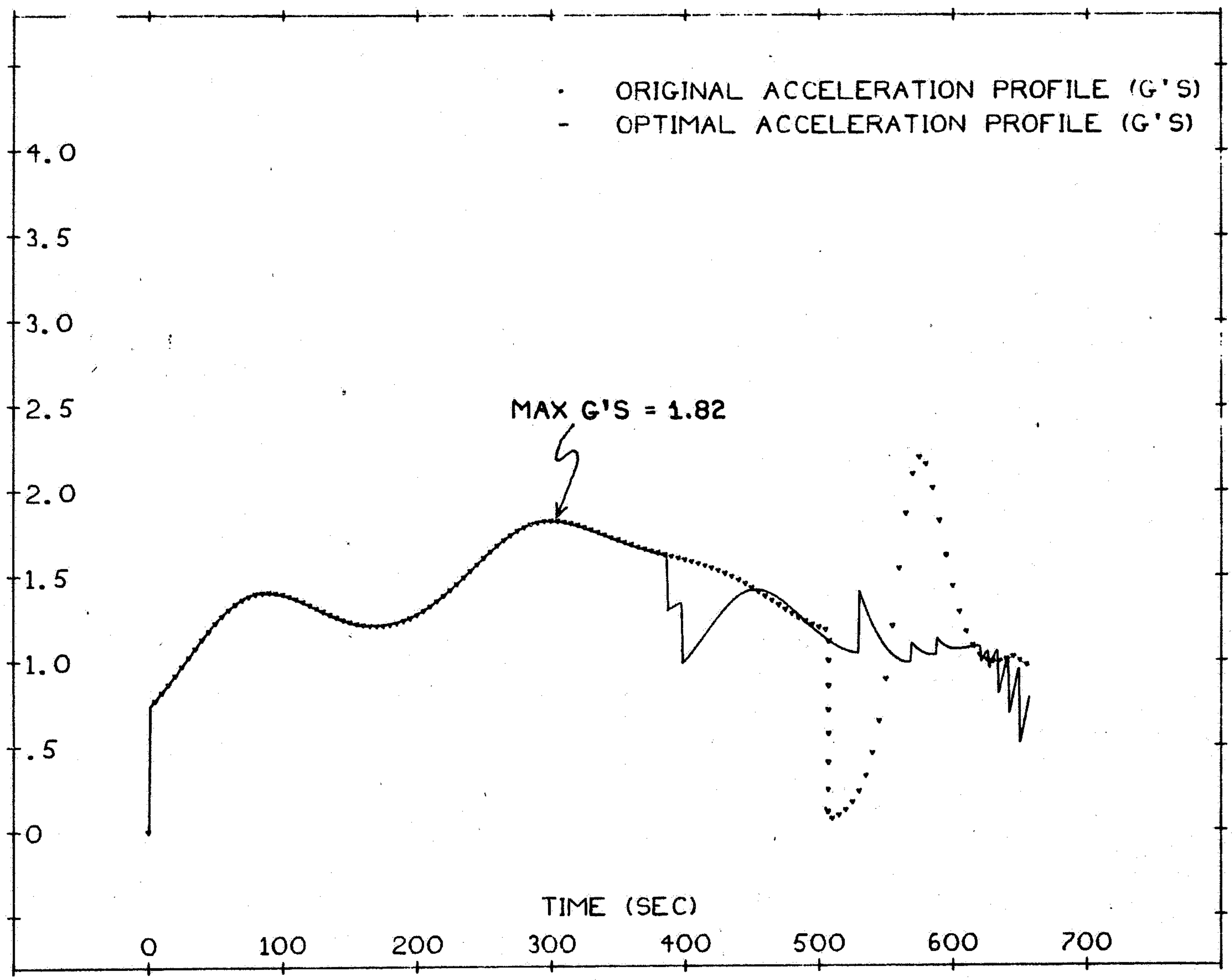
ORIGINAL ROLL ANGLE (DEG)
OPTIMAL ROLL ANGLE (DEG)
180
150
120
90
60
30
0
TIME (SEC)
0
100
200
300
400
500
600
700

Figure 6.2.7d



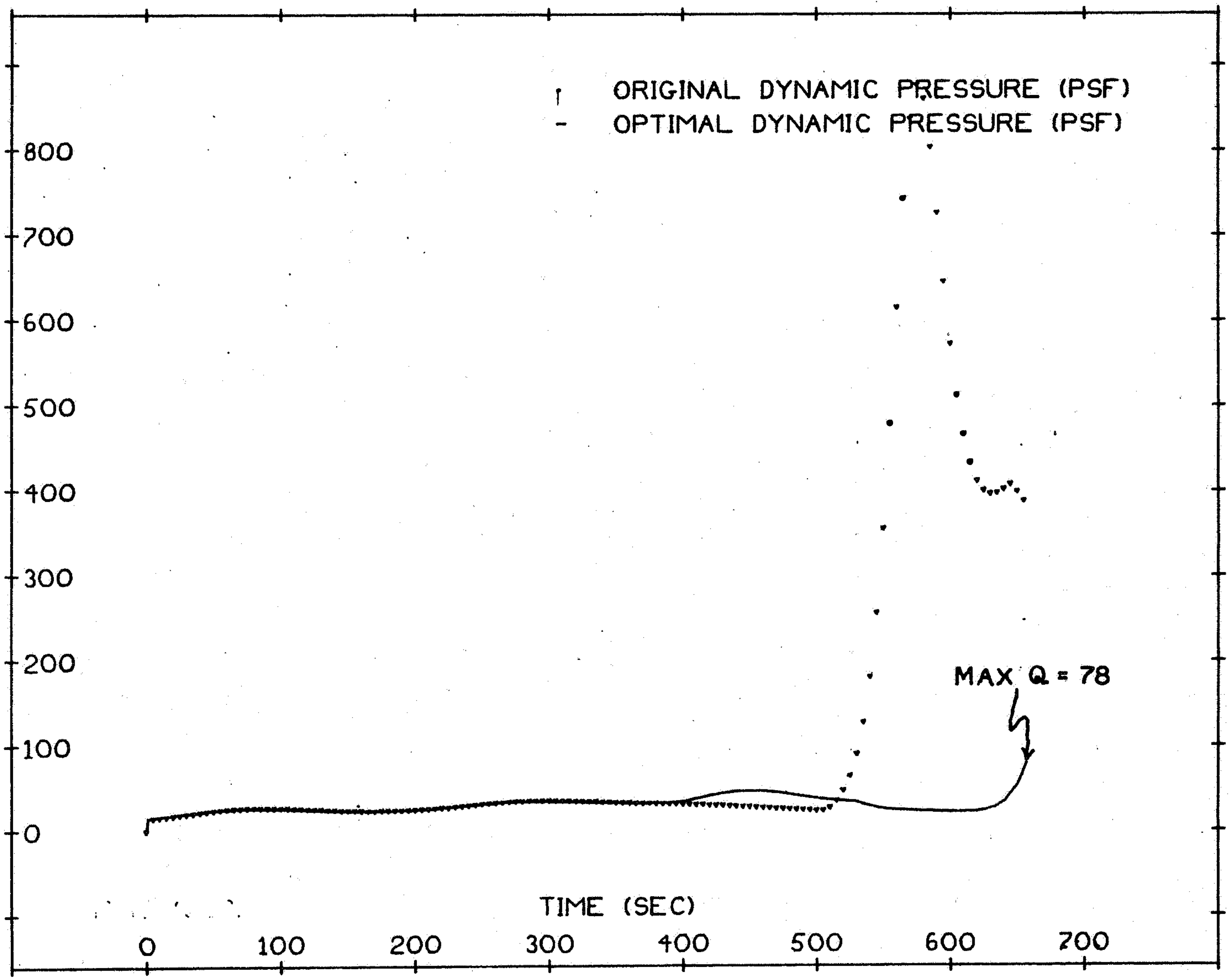
|A| HISTORY
η = 2.5
∂ (t0) = -0.228
2.5
2.0
1.5
1.0
.5
0
TIME (SEC)
700
600
500
400
300
200
100
0

Figure 6.2.7e



· ORIGINAL ACCELERATION PROFILE (G'S)
- OPTIMAL ACCELERATION PROFILE (G'S)
MAX G'S = 1.82
4.0
3.5
3.0
2.5
2.0
1.5
1.0
.5
0
TIME (SEC)
0
100
200
300
400
500
600
700

Figure 6.2.7f



ORIGINAL DYNAMIC PRESSURE (PSF)
OPTIMAL DYNAMIC PRESSURE (PSF)
800
700
600
500
400
300
200
100
0
MAX Q = 78
TIME (SEC)
0
100
200
300
400
500
600
700

Figure 6.2.7 g

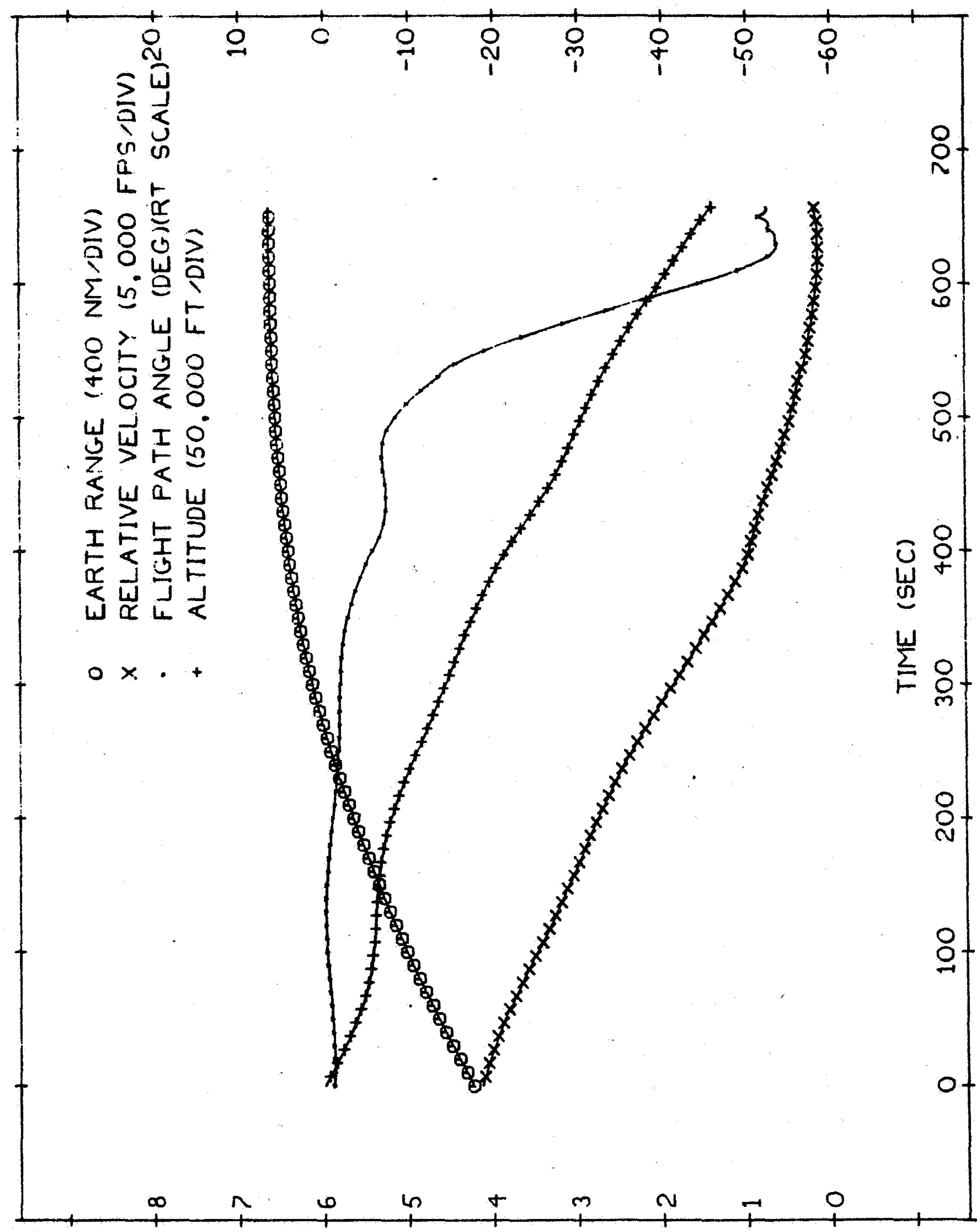

o EARTH RANGE (400 NM/DIV)
x RELATIVE VELOCITY (5,000 FPS/DIV)
· FLIGHT PATH ANGLE (DEG)(RT SCALE)
+ ALTITUDE (50,000 FT/DIV)
TIME (SEC)
0
100
200
300
400
500
600
700
0
1
2
3
4
5
6
7
8
20
10
0
-10
-20
-30
-40
-50
-60

Figure 6. 2. 7 h

In the fourth, fifth, and seventh cases, the shortened time interval did not totally eliminate the instability points. Upon further detailed examination of the non-linear system, along with the formulation and its corresponding new linear system based on converting the reverse differential Eq. (4.3.3) into the form of the non-linear matrix Riccati equation, the controllability and observability in the deterministic sense of the formulation were questioned. The formulation was found to be both controllable and deterministically observable, thus reverting the cause of the existence of the instabilities to probably the propagation of numerical errors in the computations for integrating the very sensitive reverse differential equations. This coupled with the fact that practical constraints precluded the utilization of a desirably smaller reverse integration time step is adequate cause for the sometimes diverging behavior exhibited by some of the reverse variables of integration. The shortened total time interval served only to lessen the frequency of such occurences, but was incapable of completely alleviating the problem. Another tact of rescaling the values of the state variables only served to worsen the situation. The determination of the controllability and observability of the system formulation is fully contained in Appendix D.

CHAPTER VII

CONCLUSIONS

7.1 Summary

A new second order approach for determining and optimizing nominal trajectories for the reentry phase of the Space Transportation System utilizing the notion of differential dynamic programming has been formulated. A globally positive definite inverse second partial derivative matrix, H_{uu}, is no longer required. The problem of an unbounded Riccati equation is circumvented to a certain extent by the advent of a new step size adjustment method to within the numerical accuracy of the computations involved in the reverse integration of the matrix Riccati equation. Rapid convergence of the non-linear problem is retarded only by the numerical inaccuracies of the computations.

A new approach to confronting the problem of a singular H_{uu} matrix in the reverse differential equation, where $[H_{uu}]^{-1}$ is required, is introduced. It affords a truer representation by reducing the dimension of the control vector so that H_{uu} becomes a scalar, namely its (1, 1) component; and substituting this one value rather than predetermined values for the entire matrix. Essentially, the effect of the temporarily discarded control variable on the Hamilton is ignored. In addition, an essentially control constrained problem has been converted into an unconstrained control problem through use of a finite point data table. By concluding that no additional benefits in cost can be gained outside of a certain control range of interest, a search within that range constitutes a search over the entire control space with respect to the overall cost functional or performance index.

Cases were run employing the general conditions and characteristics expected for the reentry of the STS. The major problems encountered appeared to originate from the numerical inaccuracies propagated through the integrations, and from the finite representation of a continuous system by sampled data at one second intervals. The lengthiness of the overall time interval of interest served to intensify the aggravating problem of numerical integration inaccuracy, especially for the very sensitive reverse differential equations. The desirability of decreasing the reverse integration time step duration was mainly precluded by the already overburdened requirements on computation time and computer core memory allocation. For well defined optimal control problems, optimum trajectories were established and their corresponding proper control histories and influence functions were generated.

7.2 Conclusions

The novel approach of applying the notion of differential dynamic programming to determining and optimizing atmospheric reentry trajectories is impressed. With a prudent choice of weighting factors and a reasonable initial estimate of the control histories, optimal solutions may be generated. Within a local convex region of the Hamiltonian in the control space, a magnitude of the "a" variable less than a tolerance, η, for all times of interest signifies that the local optimal solution has been determined. The validity of these optimal trajectories is dependent upon the accuracy of the aerodynamic model as reflected in the numerical values for the lift and drag coefficients, and the exactness with which they and their derivatives are extracted from a finite aerodynamic data point collection.

Each forward integration searching for a feasible value for t_1 to determine a new optimized interval of time is essentially an open loop simulation or forward integration of the state equations, closed by the "nearness criterion" of the actual cost improvement derived from each forward integration as compared with the predicted cost improvement. Here, man is eliminated from the closed loop optimization analysis, and the flaws and disadvantages of a simple acceptance and rejection criterion in the "nearness" test are made imminent. The overall efficiency and effectiveness of finding an appropriate optimized interval are impaired. A perhaps more complicated "nearness criterion" is required, possibly with "c" becoming smaller as the optimal solution is closer approached so that in the end, any improvement in cost is accepted. A final value for "c" of zero is an indication of the numerical inaccuracies that must be tolerated as the performance index improvements become smaller and smaller. A routine for judging the acceptability of the nearness of $|a|$ to "c" must not, however judicious it might appear, hamper the timewise efficiency of the overall optimizing process. Obviously, a less complicated and demanding criterion should be employed during the initial iterations.

Inclusion of eight different performance criteria with a weighting pattern for penalization renders a very flexible algorithm for determining optimal trajectories for a diversity of situations. However, care must be taken in selecting an appropriate set of weighting factors. It is due to the number and variety of parameters involved that a soft-constraint quadratic penalty cost functional was adopted. A system may become over-constrained so that no solution exists that will satisfy all the desired conditions. A soft constraint approach merely allows the system an additional freedom to violate the least important cost wise constraints, so as not to impede the proper determination of an optimal solution.

7.3 Significance

Such an approach has been to afford a closed loop methodology in optimal nominal atmospheric reentry trajectory determination. Open loop guesswork is now eliminated in favor of a consistent framework from which to generate and examine control strategies. The variables of the reverse integrations are the influence functions employed as the gains in the closed loop perturbation guidance control law. Whereas in the Apollo reentry logic the gains were generated from a first order adjoint method, the gains for the STS reentry are generated from a second order differential dynamic programming procedure and based on an optimal nominal trajectory. This importance is accentuated by the inclusion of a pitch down transition to cruise maneuver which may render highly sensitive influence functions which in turn might require higher order or cross coupled terms in the closed loop control law.

7.4 Recommendations

A useful contribution to the overall effectiveness of such a differential dynamic programming approach would be to determine a synchronization and interpolation procedure to facilitate lessening the inaccuracies encountered in the integrations of the reverse differential equations. This tact would be more feasible from a pragmatic point of view than merely decreasing the duration of the reverse integration time step.

Another very useful contribution would be to effect a more pliable and reliable "nearness test" for the acceptance and rejection of possible times for t_1. A very effective criteria would aid in reducing the computation time required to determine an appropriate t_1 on each iteration as well as possibly decreasing the total number of iterations required by accepting as much cost improvement per step as is feasibly possible. In addition, the determination of the proper t_1 in the later iterations directly affects how close and how quickly the optimal solution may be attained.

A third useful addition would be the incorporation of a procedure to vary the terminal time. Although an extended algorithm does exist to this end[J-3], the added dimensionality of the formulation would entail much more burdensome computations.

The ideas of Stengel[S-1], may be used to penalize destabilizing values of angle of attack during the transonic transition maneuvers. To obtain better accuracy during the later iterations, the weighting factors may be increased with iteration number. Another alternative currently being investigated by Pu[†] is a hybrid formulation, where a more accurate classical boundary value iteration technique is utilized after differential dynamic programming no longer yields an improvement in cost.

[†]Member of the Apollo Space Guidance Analysis Division, MIT Draper Laboratory.

In addition, operational considerations and control constraints, may be introduced into the optimization process, perhaps to render a control logic least sensitive and prone to errors emanating from bad information. Lastly, improved methods are desirable in the extracting of aerodynamic coefficients and their derivatives from three dimensional finite data point tables.

APPENDIX A

A USEFUL FORM OF THE HAMILTON-JACOBI-BELLMAN PARTIAL DIFFERENTIAL EQUATION

We write the Hamilton-Jacobi-Bellman partial differential equation (3.1.22) in terms of the nominal trajectory according to equations (3.1.17) - (3.1.18):

$$-\frac{\partial J^o}{\partial t}(\underline{\bar{x}}(t) + \delta\underline{x}(t),\ t) = \min_{\delta\underline{u}(t)} \Big\{ L[\underline{\bar{x}}(t) + \delta\underline{x}(t),\ \underline{\bar{u}}(t) + \delta\underline{u}(t),\ t]$$

$$+ \frac{\partial J^o}{\partial \underline{x}}(\underline{\bar{x}}(t) + \delta\underline{x}(t),\ t)\, \underline{f}\,[\underline{\bar{x}}(t) + \delta\underline{x}(t),\ \underline{\bar{u}}(t) + \delta\underline{u}(t),\ t] \Big\} \qquad \text{(A.1)}$$

Equation (A.1) is exact, as no approximations have been introduced yet. If the performance index is sufficiently smooth to permit a power series expansion in $\delta\underline{x}(t)$ about $\underline{\bar{x}}(t)$, there follows:

$$J^o(\underline{\bar{x}}(t) + \delta\underline{x}(t),\ t) = J^o(\underline{\bar{x}}(t),\ t) +$$

$$J^o_{\underline{x}}\Big]_{\underline{\bar{x}}(t)} \delta\underline{x}(t) + 1/2\ \delta\underline{x}(t)^T\, J^o_{\underline{xx}}\Big]_{\underline{\bar{x}}(t)} \delta\underline{x}(t)$$

$$+ \text{ higher order terms} \qquad \text{(A.2)}$$

If we rewrite the optimal cost according to equation (3.1.23) as

$$J^o(\underline{\bar{x}}(t),\ t) = \bar{J}(\underline{\bar{x}}(t),\ t) + a^o(\underline{\bar{x}}(t),\ t) \qquad \text{(A.3)}$$

where the controls are specified by equations (3.1.24) - (3.1.25), equation (A.2) becomes

$$J^o(\underline{\bar{x}}(t) + \delta\underline{x}(t)) = \bar{J}(\underline{\bar{x}}(t), t) + a^o(\underline{\bar{x}}(t), t)$$

$$+ J^o_{\underline{x}}\Big]_{\underline{\bar{x}}(t)} \delta\underline{x}(t) + 1/2\ \delta\underline{x}(t)^T J^o_{\underline{xx}}\Big]_{\underline{\bar{x}}(t)} \delta\underline{x}(t)$$

$$+ \text{ higher order terms} \qquad \text{(A. 4)}$$

Equation (A. 4) transforms equation (A. 1) into:

$$-\frac{\partial\bar{J}}{\partial t} - \frac{\partial}{\partial t}\left\{a^o(\underline{\bar{x}}(t), t)\right\} - \frac{\partial}{\partial t}\left[J^o_{\underline{x}}\right]_{\underline{\bar{x}}(t)} \delta\underline{x}(t)$$

$$- 1/2\ \delta\underline{x}(t)^T \frac{\partial}{\partial t}\left[J^o_{\underline{xx}}\right]_{\underline{\bar{x}}(t)} \delta\underline{x}(t) + \text{higher order terms}$$

$$= \min_{\delta\underline{u}(t)} \Big\{L\left[\underline{\bar{x}}(t) + \delta\underline{x}(t), \underline{\bar{u}}(t) + \delta\underline{u}(t), t\right]$$

$$+ \left(J^o_{\underline{x}}\Big]_{\underline{\bar{x}}(t)} + J^o_{\underline{xx}}\Big]_{\underline{\bar{x}}(t)} \delta\underline{x}(t) + \text{higher order terms}\right)$$

$$\underline{f}\left[\underline{\bar{x}}(t) + \delta\underline{x}(t), \underline{\bar{u}}(t) + \delta\underline{u}(t), t\right]\Big\} \qquad \text{(A.5)}$$

Confining the new perturbed trajectory to the neighborhood of the nominal trajectory ensures a small enough $\delta\underline{x}(t)$ so that higher order terms in $\delta\underline{x}(t)$ may be neglected. A sufficiently small $\delta\underline{x}(t)$ further implies that a quadratic expansion is an adequate representation of the optimal performance index in the neighborhood of the nominal trajectory. Utilizing these hypotheses, equation (A.5) is reexpressed thusly:

$$-\frac{\partial \bar{J}}{\partial t} - \frac{\partial}{\partial t}\left\{a\left(\underline{\bar{x}}(t),\ t\right)\right\} - \frac{\partial}{\partial t}\left[J_{\underline{x}}\right]_{\underline{\bar{x}}(t)} \delta\underline{x}(t)$$

$$-1/2\ \delta\underline{x}(t)^T \frac{\partial}{\partial t}\left[J_{\underline{\underline{xx}}}\right]_{\underline{\bar{x}}(t)} \delta\underline{x}(t)$$

$$\cong \min_{\delta\underline{u}(t)} \left\{ L\left[\underline{\bar{x}}(t) + \delta\underline{x}(t),\ \underline{\bar{u}}(t) + \delta\underline{u}(t),\ t\right]\right.$$

$$\left. + \left(J_{\underline{x}}\Big]_{\underline{\bar{x}}(t)} + J_{\underline{\underline{xx}}}\Big]_{\underline{\bar{x}}(t)} \delta\underline{x}(t)\right) \underline{f}\left[\underline{\bar{x}}(t) + \delta\underline{x}(t),\ \underline{\bar{u}}(t) + \delta\underline{u}(t),\ t\right]\right\} \qquad (A.6)$$

This is the form of equation (3.1.26). Note that the superscript o has been dropped for the reasons noted at the end of Section 3.1. Additional relations are given for the sake of completeness:

$$\frac{\partial \bar{J}}{\partial t} = \frac{\partial}{\partial t} \bar{J}\left(\underline{\bar{x}}(t),\ t\right) \qquad (A.7)$$

$$J\left(\underline{\bar{x}}(t) + \delta\underline{x}(t),\ t\right) = \bar{J}\left(\underline{\bar{x}}(t),\ t\right)$$

$$+ a\left(\underline{\bar{x}}(t),\ t\right) + J_{\underline{x}}\Big]_{\underline{\bar{x}}(t)} \delta\underline{x}(t) + 1/2\ \delta\underline{x}(t)^T\ J_{\underline{\underline{xx}}}\Big]_{\underline{\bar{x}}(t)} \delta\underline{x}(t) \qquad (A.8)$$

$$J_{\underline{x}}\left(\underline{\bar{x}}It) + \delta\underline{x}(t),\ t\right) = J_{\underline{x}}\left(\underline{\bar{x}}(t),\ t\right) + J_{\underline{\underline{xx}}}\left(\underline{\bar{x}}(t),\ t\right) \delta\underline{x}(t) \qquad (A.9)$$

where $J_{\underline{x}}\Big]_{\underline{\bar{x}}(t)} = J_{\underline{x}}\left(\underline{\bar{x}}(t),\ t\right)$ and $J_{\underline{\underline{xx}}}\Big]_{\underline{\bar{x}}(t)} = J_{\underline{\underline{xx}}}\left(\underline{\bar{x}}(t),\ t\right)$.

APPENDIX B

AN ANALYTICAL DERIVATION OF THE FORM OF THE REVERSE DIFFERENTIAL EQUATIONS

We take equation (3.1.26) or equivalently equation (A.6):

$$- \frac{\partial \overline{J}}{\partial t} - \frac{\partial}{\partial t} \left\{ a \left(\overline{\underline{x}}(t), \ t \right) \right\} - \frac{\partial}{\partial t} \left[J_{\underline{x}} \right]_{\overline{\underline{x}}(t)} \delta \underline{x}(t) - \frac{1}{2} \, \delta \underline{x}(t)^T \frac{\partial}{\partial t} \left[J_{\underline{xx}} \right]_{\overline{\underline{x}}(t)} \delta \underline{x}(t)$$

$$\cong \min_{\delta \underline{u}(t)} \Big\{ L \left[\overline{\underline{x}}(t) + \delta \underline{x}(t), \ \overline{\underline{u}}(t) + \delta \underline{u}(t), \ t \right] + \Big(J_{\underline{x}} \Big]_{\overline{\underline{x}}(t)} + J_{\underline{x}\underline{x}} \Big]_{\overline{\underline{x}}(t)} \delta \underline{x}(t) \Big)$$

$$\underline{f} \left[\overline{\underline{x}}(t) + \delta \underline{x}(t), \ \overline{\underline{u}}(t) + \delta \underline{u}(t), \ t \right] \Big\} \tag{B.1}$$

Setting $\delta \underline{x}(t) = 0$, the right hand side of equation (B.1) becomes:

$$\min_{\delta \underline{u}(t)} \Big\{ L \left[\overline{\underline{x}}(t), \ \overline{\underline{u}}(t) + \delta \underline{u}(t), \ t \right] + J_{\underline{x}} \Big]_{\overline{\underline{x}}(t)} \underline{f} \left[\overline{\underline{x}}(t), \ \overline{\underline{u}}(t) + \delta \underline{u}(t), \ t \right] \Big\} \tag{B.2}$$

Minimizing the expression of equation (B.2) according to the control of equation (3.2.1) gives

$$L \left[\overline{\underline{x}}(t), \ \underline{u}^*(t), \ t \right] + J_{\underline{x}} \Big]_{\overline{\underline{x}}(t)} \underline{f} \left[\overline{\underline{x}}(t), \ \underline{u}^*(t), \ t \right] \tag{B.3}$$

Now reintroduce variations $\delta \underline{x}(t)$ about $\overline{\underline{x}}(t)$. In order to maintain minimality of the right hand side of equation (B.1), we must also reintroduce $\delta \underline{u}(t)$, but now referenced about $\underline{u}^*(t)$ according to equation (3.2.5):

$$- \frac{\partial \bar{J}}{\partial t} - \frac{\partial}{\partial t}\left\{a\,(\underline{\bar{x}}(t),\ t)\right\} - \frac{\partial}{\partial t}\left[J_{\underline{x}}\right]_{\underline{\bar{x}}(t)} \delta\underline{x}(t) - \frac{1}{2}\,\delta\underline{x}(t)^T \frac{\partial}{\partial t}\left[J_{\underline{x}\,\underline{x}}\right]_{\underline{\bar{x}}(t)} \delta\underline{x}(t)$$

$$= \min_{\delta\underline{u}(t)} \left\{ H\left(\underline{\bar{x}}(t) + \delta\underline{x}(t),\ \underline{u}^*(t) + \delta\underline{u}(t),\ J_{\underline{x}}\Big]_{\underline{\bar{x}}(t)} + J_{\underline{x}\,\underline{x}}\Big]_{\underline{\bar{x}}(t)} \delta\underline{x}(t),\ t\right)\right\} \tag{B.4}$$

and utilizing the definition of the Hamiltonian as given by equation (3.1.14). This is equation (3.2.2). Expansion of the right hand side of equation (B.4) about $\underline{\bar{x}}(t)$ and $\underline{u}^*(t)$ according to equations (3.2.4) and (3.2.5) leads to

$$\min_{\delta\underline{u}(t)} \left\{ H + H_{\underline{u}}\ \delta\underline{u}(t) + H_{\underline{x}}\ \delta\underline{x}(t) + \left[J_{\underline{x}\,\underline{x}}\,\underline{f}\right]^T \delta\underline{x}(t) \right.$$

$$+ \delta\underline{u}(t)^T \left(H_{\underline{u}\,\underline{x}} + \underline{f}_{\underline{u}}^T\ J_{\underline{x}\,\underline{x}}\right) \delta\underline{x}(t) + \frac{1}{2}\,\delta\underline{u}(t)^T\ H_{\underline{u}\,\underline{u}}\ \delta\underline{u}(t)$$

$$\left. + \frac{1}{2}\,\delta\underline{x}(t)^T \left(H_{\underline{x}\,\underline{x}} + \underline{f}_{\underline{x}}^T\ J_{\underline{x}\,\underline{x}} + J_{\underline{x}\,\underline{x}}\,\underline{f}_{\underline{x}}\right) \delta\underline{x}(t) + \text{higher order terms}\right\} \tag{B.5}$$

where all quantities are evaluated at $[\underline{\bar{x}}(t),\ \underline{u}^*(t),\ t]$. We desire an optimal linear feedback controller of the form stated in equation (3.2.6):

$$\delta\underline{u}(t) = \beta(t)\delta\underline{x}(t) \tag{B.6}$$

where $\beta(t)$ is selected to maintain the minimality of equation (B.5) To this end, we differentiate equation (B.5) with respect to $\delta\underline{u}(t)$:

$$H_{\underline{u}} + H_{\underline{u}\,\underline{u}}\ \beta(t)\ \delta\underline{x}(t) + \left(H_{\underline{u}\,\underline{x}} + \underline{f}_{\underline{u}}^T\ J_{\underline{x}\,\underline{x}}\right)\delta\underline{x}(t) + \text{higher order terms} = 0 \tag{B.7}$$

with all terms evaluated at $[\underline{\bar{x}}(t),\ \underline{u}^*(t),\ t]$. This vanished because a properly chosen $\beta(t)$ maintains the necessary condition of optimality as mentioned before. Since equation (3.2.3) states that $H_{\underline{u}} = 0$, and if $\delta\underline{x}(t)$ is sufficiently small, the coefficient of the first order term in $\delta\underline{x}(t)$ of equation (B.7) may be equated to zero to yield the expression for the optimal linear feedback controller as given by equation (3.2.8):

$$\beta(t) = -\left[H_{\underline{u}\,\underline{u}}\right]^{-1} \left(H_{\underline{u}\,\underline{x}} + \underline{f}_{\underline{u}}^T\ J_{\underline{x}\,\underline{x}}\right) \tag{B.8}$$

Here again all terms are evaluated at $[\bar{\underline{x}}(t), \underline{u}^*(t), t]$. If equation (B.6) is substituted into equation (B.5), and we retain only those terms up to second order in $\delta\underline{x}(t)$, we obtain:

$$H + \left(H_{\underline{x}} + J_{\underline{x}\underline{x}}\underline{f} + \beta(t)^T H_{\underline{u}}\right)\delta\underline{x}(t)$$

$$+\frac{1}{2}\delta\underline{x}(t)^T\left(H_{\underline{x}\underline{x}} + \underline{f}_{\underline{x}}^T J_{\underline{x}\underline{x}} + J_{\underline{x}\underline{x}}\underline{f}_{\underline{x}} - \beta(t)^T H_{\underline{u}\underline{u}}\beta(t)\right)\delta\underline{x}(t) \tag{B.9}$$

Equation (B.9) is merely the right hand side of equation (B.4), allowing us to equate the coefficients of like powers of $\delta\underline{x}(t)$ to obtain equations (3.2.9) - (3.2.11):

$$-\frac{\partial\bar{J}}{\partial t} - \frac{\partial a}{\partial t} = H \tag{B.10}$$

$$-\frac{\partial}{\partial t}\left(J_{\underline{x}}\right) = H_{\underline{x}} + J_{\underline{x}\underline{x}}\underline{f} \tag{B.11}$$

$$-\frac{\partial}{\partial t}\left(J_{\underline{x}\underline{x}}\right) = H_{\underline{x}\underline{x}} + \underline{f}_{\underline{x}}^T J_{\underline{x}\underline{x}} + J_{\underline{x}\underline{x}} f_{\underline{x}}$$

$$-\left(H_{\underline{u}\underline{x}} + \underline{f}_{\underline{u}}^T J_{\underline{x}\underline{x}}\right)^T\left[H_{\underline{u}\underline{u}}\right]^{-1}\left(H_{\underline{u}\underline{x}} + \underline{f}_{\underline{u}}^T J_{\underline{x}\underline{x}}\right) \tag{B.12}$$

where all terms are evaluated at $[\bar{\underline{x}}(t), u^*(t), t]$. $\bar{J}$, a, $J_{\underline{x}}$, and $J_{\underline{x}\underline{x}}$ are functions of the nominal trajectory, permitting the following full differential expressions:

$$\frac{d}{dt}\left\{\bar{J}(\bar{\underline{x}}(t), t) + a(\bar{\underline{x}}(t), t)\right\} = \frac{\partial}{\partial t}\left\{\bar{J}(\bar{\underline{x}}(t), t) + a(\bar{\underline{x}}(t), t)\right\}$$

$$+ J_{\underline{x}}(\bar{\underline{x}}(t), t)\,\underline{f}(\bar{\underline{x}}(t), \bar{\underline{u}}(t), t) \tag{B.13}$$

$$\frac{d}{dt}\left\{J_{\underline{x}}(\overline{\underline{x}}(t), t)\right\} = \frac{\partial}{\partial t}\left\{J_{\underline{x}}(\overline{\underline{x}}(t), t)\right\} + J_{\underline{x}\underline{x}}(\overline{\underline{x}}(t), t)\, \underline{f}(\overline{\underline{x}}(t), \overline{\underline{u}}(t), t) \tag{B.14}$$

$$\frac{d}{dt}\left\{J_{\underline{x}\underline{x}}(\overline{\underline{x}}(t), t)\right\} = \frac{\partial}{\partial t}\left\{J_{\underline{x}\underline{x}}(\overline{\underline{x}}(t), t)\right\} \tag{B.15}$$

for sufficiently small $\delta \underline{x}(t)$. We also note that

$$-\frac{d}{dt}\left\{J(\overline{\underline{x}}(t), t)\right\} = L(\overline{\underline{x}}(t), \overline{\underline{u}}(t), t) \tag{B.16}$$

where use has been made of the formula for the differential of an integral with t as the lower limit of integration. Substituting equations (B.13) - (B.16) into equations (B.10) - (B.12) yields the desired reverse differential equations (3.2.12) - (3.2.14):

$$-\dot{a} = H - H\left(\overline{\underline{x}}(t), \overline{\underline{u}}(t), J_{\underline{x}}, t\right) \tag{B.17}$$

$$-\dot{J}_{\underline{x}} = H_{\underline{x}} + J_{\underline{x}\underline{x}}\left(\underline{f} - \underline{f}(\overline{\underline{x}}(t), \overline{\underline{u}}(t), t)\right) \tag{B.18}$$

$$-\dot{J}_{\underline{x}\underline{x}} = H_{\underline{x}\underline{x}} + \underline{f}_{\underline{x}}^{T} J_{\underline{x}\underline{x}} + J_{\underline{x}\underline{x}}\underline{f}_{\underline{x}} - \left(H_{\underline{u}\underline{x}} + \underline{f}_{\underline{u}}^{T} J_{\underline{x}\underline{x}}\right)^{T}\left[H_{\underline{u}\underline{u}}\right]^{-1}\left(H_{\underline{u}\underline{x}} + \underline{f}_{\underline{u}}^{T} J_{\underline{x}\underline{x}}\right) \tag{B.19}$$

where all quantities are evaluated at $[\overline{\underline{x}}(t), \underline{u}^{*}(t), t]$ unless otherwise indicated. The dot notation indicates the total differential operation with respect to time. Boundary conditions for equations (B.17) - (B.19) are as given in equations (3.2.15) - (3.2.17), and are stated here for the sake of completeness:

$$a(t_f) = 0 \tag{B.20}$$

$$J_{\underline{x}}\left(t_f\right) = F_{\underline{x}}\left(\overline{\underline{x}}(t_f), t_f\right) \tag{B.21}$$

$$J_{\underline{x}\underline{x}}\left(t_f\right) = F_{\underline{x}\underline{x}}\left(\overline{\underline{x}}\left(t_f\right), t_f\right) \tag{B.22}$$

APPENDIX C

FORMULATION OF THE REVERSE DIFFERENTIAL EQUATIONS AND THEIR TERMINAL BOUNDARY CONDITIONS

The form of the Hamiltonian stated in equation (4.3.4) is

$$H\left[\underline{x}(t),\ \underline{u}(t),\ \frac{\partial J^o}{\partial \underline{x}},\ t\right] = L\left[\underline{x}(t),\ \underline{u}(t),\ t\right] + \frac{\partial J^o}{\partial \underline{x}}\ \underline{f}\left[\underline{x}(t),\ \underline{u}(t),\ t\right] \tag{C.1}$$

Substitution of equations (4.1.1) - (4.1.4), or (4.1.12), and (4.2.2) into this expression yields

$$H = \frac{1}{2}\begin{bmatrix} \delta\dot{q}_c \\ \delta g's \\ \delta q \\ \tilde{\delta}\dot{q}_c \end{bmatrix}^T \begin{bmatrix} W\dot{q}_c & 0 & 0 & 0 \\ 0 & Wg's & 0 & 0 \\ 0 & 0 & Wq & 0 \\ 0 & 0 & 0 & 2\sqrt{W\dot{q}_c} \end{bmatrix} \begin{bmatrix} \delta\dot{q}_c \\ \delta g's \\ \delta q \\ \tilde{\delta}\dot{q}_c \end{bmatrix}$$

$$+\ J_{\underline{x}} \begin{bmatrix} v\cos\gamma \dfrac{1}{1 + h/R_e} \\ \dfrac{-q}{W/S}\, g\, C_D + \left(-g + \dfrac{v^2}{R_e + h}\right)\sin\gamma \\ \dfrac{q/v}{W/S}\, g\, C_L + \left(-g/v + \dfrac{v}{R_e + h}\right)\cos\gamma \\ v\sin\gamma \end{bmatrix} \tag{C.2}$$

The partial differential of $\underline{f}$ ($\underline{x}$(t), $\underline{u}$(t), t) of equation (4.1.12) with respect to the state vector given by equation (4.1.10) is expressed as:

$$\underline{f}_{\underline{x}} = \begin{bmatrix} (\dot{x}_r)_{x_r} & (\dot{x}_r)_v & (\dot{x}_r)_\gamma & (\dot{x}_r)_h \\ (\dot{v})_{x_r} & (\dot{v})_v & (\dot{v})_\gamma & (\dot{v})_h \\ (\dot{\gamma})_{x_r} & (\dot{\gamma})_v & (\dot{\gamma})_\gamma & (\dot{\gamma})_h \\ (\dot{h})_{x_r} & (\dot{h})_v & (\dot{h})_\gamma & (\dot{h})_h \end{bmatrix} \tag{C.3}$$

where

$$(\dot{x}_r)_{x_r} = 0 \tag{C.4}$$

$$(\dot{x}_r)_v = \cos\gamma \frac{1}{1 + h/R_e} \tag{C.5}$$

$$(\dot{x}_r)_\gamma = -V \sin\gamma \frac{1}{1 + h/R_e} \tag{C.6}$$

$$(\dot{x}_r)_h = -V \cos\gamma \frac{R_e}{(R_e + h)^2} \tag{C.7}$$

$$(\dot{v})_{x_r} = 0 \tag{C.8}$$

$$(\dot{v})_v = \frac{-\rho^v}{W/S} g\, C_D - \frac{q}{W/S} g\, (C_D)_v + \frac{2\, v}{R_e + h} \sin\gamma \tag{C.9}$$

$$(\dot{v})_\gamma = -\, g \cos\gamma + \frac{v^2}{R_e + h} \cos\gamma \tag{C.10}$$

$$(\dot{v})_h = -\frac{1}{2}\frac{\partial \rho}{\partial h}\frac{v^2}{W/S} g\, C_D - \frac{q}{W/S} g\, (C_D)_h - \frac{v^2}{(R_e + h)^2} \sin\gamma \quad (C.11)$$

$$(\dot{\gamma})_{x_r} = 0 \quad (C.12)$$

$$(\dot{\gamma})_v = \frac{\frac{1}{2}\rho}{W/S} g\, C_L + \frac{q/v}{W/S} g\, (C_L)_v$$

$$+ \cos\gamma \frac{1}{R_e + h} + \frac{g}{v^2}\cos\gamma \quad (C.13)$$

$$(\dot{\gamma})_\gamma = -\frac{v}{R_e + h}\sin\gamma + \frac{g}{v}\sin\gamma \quad (C.14)$$

$$(\dot{\gamma})_h = \frac{1}{2}\frac{\partial \rho}{\partial h}\frac{v}{W/S} g\, C_L + \frac{q/v}{W/S} g\, (C_L)_h$$

$$- \frac{v}{(R_e + h)^2}\cos\gamma \quad (C.15)$$

$$(\dot{h})_{x_r} = 0 \quad (C.16)$$

$$(\dot{h})_v = \sin\gamma \quad (C.17)$$

$$(\dot{h})_\gamma = v\cos\gamma \quad (C.18)$$

$$(\dot{h})_h = 0 \quad (C.19)$$

The partial differential of $\underline{f}\ (\underline{x}(t),\ \underline{u}(t),\ t)$ of equation (4.1.12) with respect to the control vector in equation (4.1.11) is

$$\underline{f_u} = \begin{bmatrix} (\dot{x}_r)_\varphi & (\dot{x}_r)_\alpha \\ (\dot{v})_\varphi & (\dot{v})_\alpha \\ (\dot{\gamma})_\varphi & (\dot{\gamma})_\alpha \\ (\dot{h})_\varphi & (\dot{h})_\alpha \end{bmatrix} \tag{C.20}$$

where

$$(\dot{x}_r)_\varphi = 0 \tag{C.21}$$

$$(\dot{x}_r)_\alpha = 0 \tag{C.22}$$

$$(\dot{v})_\varphi = 0 \tag{C.23}$$

$$(\dot{v})_\alpha = -\frac{q}{W/S}\, g\, (C_D)_\alpha \tag{C.24}$$

$$(\dot{\gamma})_\varphi = \frac{q/v}{W/S}\, g\, (C_L)_\varphi \tag{C.25}$$

$$(\dot{\gamma})_\alpha = \frac{q/v}{W/S}\, g\, (C_L)_\alpha \tag{C.26}$$

$$(\dot{h})_\varphi = 0 \tag{C.27}$$

$$(\dot{h})_\alpha = 0 \tag{C.28}$$

The forms of the partial derivatives of the Hamiltonian will now be derived starting from its basic definition:

$$H = L + J_{\underline{x}}\, \underline{f} \tag{C.29}$$

Thus

$$
\begin{aligned}
H_{\underline{x}} &= L_{\underline{x}} + \left(J_{\underline{x}}\underline{f}\right)_{\underline{x}} \\
&= L_{\underline{x}} + J_{\underline{x}}\underline{f}_{\underline{x}}
\end{aligned}
\tag{C.30}
$$

$$
\begin{aligned}
H_{\underline{\underline{xx}}} &= L_{\underline{\underline{xx}}} + \left([J_{\underline{x}}\underline{f}_{\underline{x}}]^T\right)_{\underline{x}} \\
&= L_{\underline{\underline{xx}}} + \begin{bmatrix} \begin{pmatrix} \underline{f}_{x_r}^{\;T} \\ \underline{f}_{v}^{\;T} \\ \underline{f}_{\gamma}^{\;T} \\ \underline{f}_{h}^{\;T} \end{pmatrix} J_{\underline{x}}^{\;T} \end{bmatrix}_{\underline{x}}
= L_{\underline{\underline{xx}}} + \begin{bmatrix} J_{\underline{x}}\underline{f}_{x_r} \\ J_{\underline{x}}\underline{f}_{v} \\ J_{\underline{x}}\underline{f}_{\gamma} \\ J_{\underline{x}}\underline{f}_{h} \end{bmatrix}_{\underline{x}} \\
&= L_{\underline{\underline{xx}}} + \begin{bmatrix} J_{\underline{x}}\underline{f}_{x_r\underline{x}} \\ J_{\underline{x}}\underline{f}_{v\underline{x}} \\ J_{\underline{x}}\underline{f}_{\gamma\underline{x}} \\ J_{\underline{x}}\underline{f}_{h\underline{x}} \end{bmatrix}
= L_{\underline{\underline{xx}}} + \begin{bmatrix} J_{\underline{x}}\underline{f}_{\underline{x}x_r} \\ J_{\underline{x}}\underline{f}_{\underline{x}v} \\ J_{\underline{x}}\underline{f}_{\underline{x}\gamma} \\ J_{\underline{x}}\underline{f}_{\underline{x}h} \end{bmatrix}
\end{aligned}
\tag{C.31}
$$

$$
\begin{aligned}
H_{\underline{u}} &= L_{\underline{u}} + \left(J_{\underline{x}}\underline{f}\right)_{\underline{u}} \\
&= L_{\underline{u}} + J_{\underline{x}}\underline{f}_{\underline{u}}
\end{aligned}
\tag{C.32}
$$

$$
\begin{aligned}
H_{\underline{u}\underline{u}} &= L_{\underline{u}\underline{u}} + \left(\left[J_{\underline{x}}\underline{f}_{\underline{u}}\right]^{T}\right)_{\underline{u}} \\
&= L_{\underline{u}\underline{u}} + \left[\begin{pmatrix} \underline{f}_{\phi}{}^{T} \\ \underline{f}_{\alpha}{}^{T} \end{pmatrix} J_{\underline{x}}{}^{T}\right]_{\underline{u}} \qquad = L_{\underline{u}\underline{u}} + \begin{bmatrix} J_{\underline{x}}\underline{f}_{\phi} \\ J_{\underline{x}}\underline{f}_{\alpha} \end{bmatrix}_{\underline{u}} \\
&= L_{\underline{u}\underline{u}} + \begin{bmatrix} J_{\underline{x}}\underline{f}_{\phi\underline{u}} \\ J_{\underline{x}}\underline{f}_{\alpha\underline{u}} \end{bmatrix} \qquad = L_{\underline{u}\underline{u}} + \begin{bmatrix} J_{\underline{x}}\underline{f}_{\underline{u}\phi} \\ J_{\underline{x}}\underline{f}_{\underline{u}\alpha} \end{bmatrix}
\end{aligned}
\tag{C.33}
$$

$$
\begin{aligned}
H_{\underline{u}\underline{x}} &= \left(\left[H_{\underline{u}}\right]^{T}\right)_{\underline{x}} \\
&= L_{\underline{u}\underline{x}} + \left(\left[J_{\underline{x}}\underline{f}_{\underline{u}}\right]^{T}\right)_{\underline{x}} \qquad = L_{\underline{u}\underline{x}} + \left[\begin{pmatrix} \underline{f}_{\phi}{}^{T} \\ \underline{f}_{\alpha}{}^{T} \end{pmatrix} J_{\underline{x}}{}^{T}\right]_{\underline{x}} \\
&= L_{\underline{u}\underline{x}} + \begin{bmatrix} J_{\underline{x}}\underline{f}_{\phi} \\ J_{\underline{x}}\underline{f}_{\alpha} \end{bmatrix}_{\underline{x}} \qquad = L_{\underline{u}\underline{x}} + \begin{bmatrix} J_{\underline{x}}\underline{f}_{\phi\underline{x}} \\ J_{\underline{x}}\underline{f}_{\alpha\underline{x}} \end{bmatrix} \\
&= L_{\underline{u}\underline{x}} + \begin{bmatrix} J_{\underline{x}}\underline{f}_{\underline{x}\phi} \\ J_{\underline{x}}\underline{f}_{\underline{x}\alpha} \end{bmatrix}
\end{aligned}
\tag{C.34}
$$

The partials of $\underline{f}$ required by equations (C.31), (C.33), and (C.34) are now defined:

$$\underline{f}_{\underline{x}\, x_r} = \begin{bmatrix} 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix} \qquad (C.35)$$

$$\underline{f}_{\underline{x}\, v} = \begin{bmatrix} (\dot{x}_r)_{x_r v} & (\dot{x}_r)_{vv} & (\dot{x}_r)_{\gamma v} & (\dot{x}_r)_{hv} \\ (\dot{v})_{x_r v} & (\dot{v})_{vv} & (\dot{v})_{\gamma v} & (\dot{v})_{hv} \\ (\dot{\gamma})_{x_r v} & (\dot{\gamma})_{vv} & (\dot{\gamma})_{\gamma v} & (\dot{\gamma})_{hv} \\ (\dot{h})_{x_r v} & (\dot{h})_{vv} & (\dot{h})_{\gamma v} & (\dot{h})_{hv} \end{bmatrix} \qquad (C.36)$$

where

$$(\dot{x}_r)_{x_r v} = 0 \qquad (C.37)$$

$$(\dot{x}_r)_{vv} = 0 \qquad (C.38)$$

$$(\dot{x}_r)_{\gamma v} = - \sin \gamma \frac{1}{1 + {}^{h}/R_e} \qquad (C.39)$$

$$(\dot{x}_r)_{hv} = - \cos \gamma \frac{R_e}{(R_e + h)^2} \qquad (C.40)$$

$$(\dot{v})_{x_r v} = 0 \qquad (C.41)$$

$$(\dot{v})_{vv} = -\frac{\rho}{W/S} g\, C_D - \frac{2\rho v}{W/S} g \left(C_D\right)_v - \frac{q}{W/S} g \left(C_D\right)_{vv} + \frac{2}{R_e + h} \sin\gamma \tag{C. 42}$$

$$(\dot{v})_{\gamma v} = \frac{2v}{R_e + h} \cos\gamma \tag{C. 43}$$

$$(\dot{v})_{hv} = -\frac{\partial\rho}{\partial h}\frac{v}{W/S} g\, C_D - \frac{1}{2}\frac{\partial\rho}{\partial h}\frac{v^2}{W/S} g \left(C_D\right)_v - \frac{\rho v}{W/S} g \left(C_D\right)_h - \frac{q}{W/S} g \left(C_D\right)_{hv} - \frac{2v}{(R_e + h)^2} \sin\gamma \tag{C. 44}$$

$$(\dot{\gamma})_{x_r v} = 0 \tag{C. 45}$$

$$(\dot{\gamma})_{vv} = \frac{\rho}{W/S} g \left(C_L\right)_v + \frac{q/v}{W/S} g \left(C_L\right)_{vv} - \frac{2g}{v^3} \cos\gamma \tag{C. 46}$$

$$(\dot{\gamma})_{\gamma v} = -\frac{1}{R_e + h} \sin\gamma - \frac{g}{v^2} \sin\gamma \tag{C. 47}$$

$$(\dot{\gamma})_{hv} = \frac{1}{2}\frac{\partial\rho}{\partial h}\frac{g}{W/S} C_L + \frac{1}{2}\frac{\partial\rho}{\partial h}\frac{v}{W/S} g \left(C_L\right)_v + \frac{1}{2}\rho \frac{g}{W/S}\left(C_L\right)_h + \frac{q/v}{W/S} g \left(C_L\right)_{hv} - \frac{1}{(R_e + h)^2} \cos\gamma \tag{C. 48}$$

$$(\dot{h})_{x_r v} = 0 \tag{C. 49}$$

$$(\dot{h})_{vv} = 0 \tag{C. 50}$$

$$(\dot{h})_{\gamma v} = \cos\gamma \tag{C.51}$$

$$(\dot{h})_{hv} = 0 \tag{C.52}$$

$$\underline{f}_{\underline{x}\gamma} = \begin{bmatrix} (\dot{x}_r)_{x_r\gamma} & (\dot{x}_r)_{v\gamma} & (\dot{x}_r)_{\gamma\gamma} & (\dot{x}_r)_{h\gamma} \\ (\dot{v})_{x_r\gamma} & (\dot{v})_{v\gamma} & (\dot{v})_{\gamma\gamma} & (\dot{v})_{h\gamma} \\ (\dot{\gamma})_{x_r\gamma} & (\dot{\gamma})_{v\gamma} & (\dot{\gamma})_{\gamma\gamma} & (\dot{\gamma})_{h\gamma} \\ (\dot{h})_{x_r\gamma} & (\dot{h})_{v\gamma} & (\dot{h})_{\gamma\gamma} & (\dot{h})_{h\gamma} \end{bmatrix} \tag{C.53}$$

where

$$(\dot{x}_r)_{x_r\gamma} = 0 \tag{C.54}$$

$$(\dot{x}_r)_{v\gamma} = -\sin\gamma \frac{1}{1 + h/R_e} \tag{C.55}$$

$$(\dot{x}_r)_{\gamma\gamma} = -V\cos\gamma \frac{1}{1 + h/R_e} \tag{C.56}$$

$$(\dot{x}_r)_{h\gamma} = V\sin\gamma \frac{R_e}{(R_e + h)^2} \tag{C.57}$$

$$(\dot{v})_{x_r\gamma} = 0 \tag{C.58}$$

$$(\dot{v})_{v\gamma} = \frac{2v}{R_e + h}\cos\gamma \tag{C.59}$$

$$(\dot{v})_{\gamma\gamma} = g \sin\gamma - \frac{v^2}{R_e + h} \sin\gamma \qquad (C.60)$$

$$(\dot{v})_{h\gamma} = -\frac{v^2}{(R_e + h)^2} \cos\gamma \qquad (C.61)$$

$$(\dot{\gamma})_{x_r\gamma} = 0 \qquad (C.62)$$

$$(\dot{\gamma})_{v\gamma} = -\frac{1}{R_e + h} \sin\gamma - \frac{g}{v^2} \sin\gamma \qquad (C.63)$$

$$(\dot{\gamma})_{\gamma\gamma} = -\frac{v}{R_e + h} \cos\gamma + \frac{g}{v} \cos\gamma \qquad (C.64)$$

$$(\dot{\gamma})_{h\gamma} = \frac{v}{(R_e + h)^2} \sin\gamma \qquad (C.65)$$

$$(\dot{h})_{x_r\gamma} = 0 \qquad (C.66)$$

$$(\dot{h})_{v\gamma} = \cos\gamma \qquad (C.67)$$

$$(\dot{h})_{\gamma\gamma} = -v \sin\gamma \qquad (C.68)$$

$$(\dot{h})_{h\gamma} = 0 \qquad (C.69)$$

$$\underline{f}_{xh} = \begin{bmatrix} (\dot{x}_r)_{x_r h} & (\dot{x}_r)_{vh} & (\dot{x}_r)_{\gamma h} & (\dot{x}_r)_{hh} \\ (\dot{v})_{x_r h} & (\dot{v})_{vh} & (\dot{v})_{\gamma h} & (\dot{v})_{hh} \\ (\dot{\gamma})_{x_r h} & (\dot{\gamma})_{vh} & (\dot{\gamma})_{\gamma h} & (\dot{\gamma})_{hh} \\ (\dot{h})_{x_r h} & (\dot{h})_{vh} & (\dot{h})_{\gamma h} & (\dot{h})_{hh} \end{bmatrix} \tag{C. 70}$$

where

$$(\dot{x}_r)_{x_r h} = 0 \tag{C. 71}$$

$$(\dot{x}_r)_{vh} = -\cos\gamma \frac{R_e}{(R_e + h)^2} \tag{C. 72}$$

$$(\dot{x}_r)_{\gamma h} = V \sin\gamma \frac{R_e}{(R_e + h)^2} \tag{C. 73}$$

$$(\dot{x}_r)_{hh} = 2 v \cos\gamma \frac{R_e}{(R_e + h)^3} \tag{C. 74}$$

$$(\dot{v})_{x_r h} = 0 \tag{C. 75}$$

$$\begin{aligned}(\dot{v})_{vh} = &- \frac{\partial\rho}{\partial h} \frac{v}{W/S} g\, C_D - \frac{\rho v}{W/S} g \left(C_D\right)_h \\ &- \frac{1}{2} \frac{\partial\rho}{\partial h} \frac{v^2}{W/S} g \left(C_D\right)_v - \frac{q}{W/S} g \left(C_D\right)_{vh} \\ &- \frac{2v}{(R_e + h)^2} \sin\gamma \end{aligned} \tag{C. 76}$$

$$(\dot{v})_{\gamma h} = \frac{-v^2}{(R_e+h)^2} \cos \gamma \qquad (C.77)$$

$$(\dot{v})_{hh} = -\frac{1}{2} \frac{\partial^2 \rho}{\partial h^2} \frac{v^2}{W/S} gC_D - \frac{\partial \rho}{\partial h} \frac{v^2}{W/S} g(C_D)_h$$

$$\frac{-q}{W/S} g(C_D)_{hh} + \frac{2v^2}{(R_e+h)^3} \sin \gamma \qquad (C.78)$$

$$(\dot{\gamma})_{x_r h} = 0 \qquad (C.79)$$

$$(\dot{\gamma})_{vh} = \frac{1}{2} \frac{\partial \rho}{\partial h} \frac{g}{W/S} C_L + \frac{1}{2}\rho \frac{g}{W/S} (C_L)_h$$

$$+ \frac{1}{2} \frac{\partial \rho}{\partial h} \frac{v}{W/S} g(C_L)_v + \frac{q/v}{W/S} g(C_L)_{vh}$$

$$- \frac{1}{(R_e+h)^2} \cos \gamma \qquad (C.80)$$

$$(\dot{\gamma})_{\gamma h} = \frac{v}{(R_e+h)^2} \sin \gamma \qquad (C.81)$$

$$(\dot{\gamma})_{hh} = \frac{1}{2} \frac{\partial^2 \rho}{\partial h^2} \frac{v}{W/S} gC_L + \frac{\partial \rho}{\partial h} \frac{v}{W/S} g(C_L)_h$$

$$+ \frac{q/v}{W/S} g(C_L)_{hh} + \frac{2v}{(R_e+h)^3} \cos \gamma \qquad (C.82)$$

$$(\dot{h})_{x_r h} = 0 \qquad (C.83)$$

$$(\dot{h})_{vh} = 0 \tag{C.84}$$

$$(\dot{h})_{\gamma h} = 0 \tag{C.85}$$

$$(\dot{h})_{hh} = 0 \tag{C.86}$$

$$\underline{f}_{\underline{u}\phi} = \begin{bmatrix} 0 & 0 \\ 0 & 0 \\ \frac{q/v}{W/S}\, g(C_L)_{\phi\phi} & \frac{q/v}{W/S}\, g(C_L)_{\phi\alpha} \\ 0 & 0 \end{bmatrix} \tag{C.87}$$

$$\underline{f}_{\underline{u}\alpha} = \begin{bmatrix} 0 & 0 \\ 0 & \frac{-q}{W/S}\, g(C_D)_{\alpha\alpha} \\ \frac{q/v}{W/S}\, g(C_L)_{\phi\alpha} & \frac{q/v}{W/S}\, g(C_L)_{\alpha\alpha} \\ 0 & 0 \end{bmatrix} \tag{C.88}$$

$$\underline{f}_{\underline{x}\phi} = \begin{bmatrix} 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & (\dot{\gamma})_{v\phi} & 0 & (\dot{\gamma})_{h\phi} \\ 0 & 0 & 0 & 0 \end{bmatrix} \tag{C.89}$$

where

$$(\dot{\gamma})_{v\phi} = \frac{1}{2}\rho \frac{g}{W/S} (C_L)_\phi + \frac{g/v}{W/S} g(C_L)_{v\phi} \tag{C.90}$$

$$(\dot{\gamma})_{h\phi} = \frac{1}{2} \frac{\partial\rho}{\partial h} \frac{v}{W/S} g(C_L)_\phi + \frac{g/v}{W/S} g(C_L)_{h\phi} \tag{C.91}$$

$$\underline{f}_{\underline{x}\alpha} = \begin{bmatrix} 0 & 0 & 0 & 0 \\ 0 & (\dot{v})_{v\alpha} & 0 & (\dot{v})_{h\alpha} \\ 0 & (\dot{\gamma})_{v\alpha} & 0 & (\dot{\gamma})_{h\alpha} \\ 0 & 0 & 0 & 0 \end{bmatrix} \tag{C.92}$$

where

$$(\dot{v})_{v\alpha} = \frac{-\rho v}{W/S} g(C_D)_\alpha - \frac{g}{W/S} g(C_D)_{v\alpha} \tag{C.93}$$

$$(\dot{v})_{h\alpha} = -\frac{1}{2} \frac{\partial\rho}{\partial h} \frac{v^2}{W/S} g(C_D)_\alpha - \frac{g}{W/S} g(C_D)_{h\alpha} \tag{C.94}$$

$$(\dot{\gamma})_{v\alpha} = \frac{1}{2}\rho \frac{g}{W/S} (C_L)_\alpha + \frac{g/v}{W/S} g(C_L)_{v\alpha} \tag{C.95}$$

$$(\dot{\gamma})_{h\alpha} = \frac{1}{2} \frac{\partial\rho}{\partial h} \frac{v}{W/S} g(C_L)_\alpha + \frac{g/v}{W/S} g(C_L)_{h\alpha} \tag{C.96}$$

The partials of the accrued cost functional L are now derived. We write equation (4.3.11):

$$L = \frac{1}{2} \{W\dot{q}_c(\delta\dot{q}_c)^2 + Wg's(\delta g's)^2 + Wq(\delta q)^2$$
$$+ 2\sqrt{W\dot{q}_c}(\sqrt{\delta\dot{q}_c})^2\} \tag{C.97}$$

$L_{\underline{x}}$ is given by equation (4.3.12) as

$$L_{\underline{x}} = \begin{bmatrix} 0 \\ \delta\dot{q}_c \; W\dot{q}_c(\delta\dot{q}_c)_v + \delta g's \; Wg's(\delta g's)_v \\ + \delta q \; Wq(\delta q)_v + \sqrt{Wq_c}(\tilde{\delta}\dot{q}_c)_v \\ 0 \\ \delta\dot{q}_c \; W\dot{q}_c(\delta\dot{q}_c)_h + \delta g's \; Wg's(\delta g's)_h \\ + \delta q \; Wq(\delta q)_h + \sqrt{Wq_c}(\tilde{\delta}\dot{q}_c)_h \end{bmatrix}^T \tag{C.98}$$

From equation (4.3.13), we proceed to obtain

$$L_{\underline{\underline{xx}}} = \begin{bmatrix} 0 & 0 & 0 & 0 \\ 0 & L_{vv} & 0 & L_{vh} \\ 0 & 0 & 0 & 0 \\ 0 & L_{hv} & 0 & L_{hh} \end{bmatrix} \tag{C.99}$$

where

$$\begin{aligned} L_{vv} = {} & W\dot{q}_c(\delta\dot{q}_c)_v^2 + \delta\dot{q}_c \; W\dot{q}_c(\delta\dot{q}_c)_{vv} \\ & + Wg's(\delta g's)_v^2 + \delta g's \; Wg's(\delta g's)_{vv} \\ & + Wq(\delta q)_v^2 + \delta q \; Wq(\delta q)_{vv} \\ & + \sqrt{Wq_c}(\tilde{\delta}\dot{q}_c)_{vv} \end{aligned} \tag{C.100}$$

$$L_{vh} = (\delta\dot{q}_c)_h \, W\dot{q}_c(\delta\dot{q}_c)_v + \delta\dot{q}_c \, W\dot{q}_c(\delta\dot{q}_c)_{vh}$$

$$+ (\delta g's)_h \, Wg's(\delta g's)_v$$

$$+ \delta g's \, Wg's(\delta g's)_{vh} + (\delta q)_h \, Wq(\delta q)_v$$

$$+ \delta q \, Wq(\delta q)_{vh} + \sqrt{Wq}_c(\tilde{\delta}\dot{q}_c)_{vh} \qquad \text{(C.101)}$$

$$L_{hv} = L_{vh} \qquad \text{(C.102)}$$

$$L_{hh} = W\dot{q}_c(\delta\dot{q}_c)_h^2 + \delta\dot{q}_c \, W\dot{q}_c(\delta\dot{q}_c)_{hh}$$

$$+ Wg's(\delta g's)_h^2 + \delta g's \, Wg's(\delta g's)_{hh}$$

$$+ Wq(\delta q)_h^2 + \delta q \, Wq(\delta q)_{hh}$$

$$+ \sqrt{Wq}_c(\tilde{\delta}\dot{q}_c)_{hh} \qquad \text{(C.103)}$$

From equation (4.3.14) we state

$$L_{\underline{u}} = \begin{bmatrix} \delta g's \; Wg's \; (\delta g's)_\phi \\ \delta g's \; Wg's \; (\delta g's)_\alpha \end{bmatrix}^T \qquad \text{(C.104)}$$

Following equation (4.3.15) we may write

$$L_{\underline{uu}} = \begin{bmatrix} L_{\phi\phi} & L_{\phi\alpha} \\ L_{\alpha\phi} & L_{\alpha\alpha} \end{bmatrix} \qquad \text{(C.105)}$$

where

$$L_{\phi\phi} = Wg's(\delta g's)_{\phi}^{2} + \delta g's\ Wg's(\delta g's)_{\phi\phi} \tag{C.106}$$

$$L_{\phi\alpha} = (\delta g's)_{\alpha}\ Wg's(\delta g's)_{\phi} + \delta g's\ Wg's(\delta g's)_{\phi\alpha} \tag{C.107}$$

$$L_{\alpha\phi} = L_{\phi\alpha} \tag{C.108}$$

$$L_{\alpha\alpha} = Wg's(\delta g's)_{\alpha}^{2} + \delta g's\ Wg's(\delta g's)_{\alpha\alpha} \tag{C.109}$$

Equation (4.3.16) leads to

$$L_{\underline{\underline{ux}}} = \begin{bmatrix} 0 & L_{\phi v} & 0 & L_{\phi h} \\ 0 & L_{\alpha v} & 0 & L_{\alpha h} \end{bmatrix} \tag{C.110}$$

where

$$L_{\phi v} = (\delta g's)_{v}\ Wg's(\delta g's)_{\phi} + \delta g's\ Wg's(\delta g's)_{v\phi} \tag{C.111}$$

$$L_{\phi h} = (\delta g's)_{h}\ Wg's(\delta g's)_{\phi} + \delta g's\ Wg's(\delta g's)_{h\phi} \tag{C.112}$$

$$L_{\alpha v} = (\delta g's)_{v}\ Wg's(\delta g's)_{\alpha} + \delta g's\ Wg's(\delta g's)_{v\alpha} \tag{C.113}$$

$$L_{\alpha h} = (\delta g's)_h \, Wg's(\delta g's)_\alpha + \delta g's \, Wg's(\delta g's)_{h\alpha} \tag{C.114}$$

The partial derivatives of the delta terms will now be given. Starting with equations (4.1.14) and (4.2.6),

$$\delta\dot{q}_c = \frac{.61433466}{\sqrt{R_N}} \left(\rho/\rho_0\right)^{0.5} \left(\frac{v}{1000}\right)^{3.15} - \dot{q}_c\Big]_{max} \tag{C.115}$$

we see that

$$(\delta\dot{q}_c)_v = 3.15 \, \dot{q}_c/v \tag{C.116}$$

$$(\delta\dot{q}_c)_{vv} = (3.15)(2.15)\dot{q}_c/v^2 \tag{C.117}$$

$$(\delta\dot{q}_c)_h = \frac{1}{2} \dot{q}_c \frac{1}{\rho} \frac{\partial\rho}{\partial h} \tag{C.118}$$

$$(\delta\dot{q}_c)_{hh} = \dot{q}_c\left(\frac{1}{2} \frac{1}{\rho} \frac{\partial^2\rho}{\partial h^2} - \frac{1}{4} \frac{1}{\rho^2} \left(\frac{\partial\rho}{\partial h}\right)^2\right) \tag{C.119}$$

$$(\delta\dot{q}_c)_{vh} = \frac{3.15}{2} \left(\dot{q}_c/v\rho\right) \frac{\partial\rho}{\partial h} \tag{C.120}$$

The partials of $\tilde{\delta}\dot{q}_c$ are identical to those for $\delta\dot{q}_c$ as given by equations (C.116) - (C.120).

Referring to equations (4.1.5) and (4.2.10) for the dynamic pressure penalty term,

$$\delta q = \frac{1}{2} \rho v^2 - q]_{max} \tag{C.121}$$

the partial derivatives are:

$$(\delta q)_v = \rho v \tag{C.122}$$

$$(\delta q)_{vv} = \rho \tag{C.123}$$

$$(\delta q)_h = \frac{1}{2} \frac{\partial \rho}{\partial h} v^2 \tag{C.124}$$

$$(\delta q)_{hh} = \frac{1}{2} \frac{\partial^2 \rho}{\partial h^2} v^2 \tag{C.125}$$

$$(\delta q)_{vh} = \frac{\partial \rho}{\partial h} v \tag{C.126}$$

From the delta terms of the acceleration given by equations (4.2.7) and (4.2.9),

$$\delta g's = \frac{q}{W/S} \left(C_D{}^2 + C_{L_{max}}{}^2 \right)^{0.5} - g's]_{max} \tag{C.127}$$

the partials are as follows:

$$(\delta g's)_v = \frac{\rho v}{W/S}\left(C_D{}^2 + C_{L_{max}}{}^2\right)^{0.5} + \frac{q}{W/S}\,\frac{C_D(C_D)_v + C_{L_{max}}\left(C_{L_{max}}\right)_v}{\left(C_D{}^2 + C_{L_{max}}{}^2\right)^{0.5}} \tag{C.128}$$

$$\begin{aligned}(\delta g's)_{vv} = {} & \frac{\rho}{W/S}\left(C_D{}^2 + C_{L_{max}}{}^2\right)^{0.5} \\ & + \frac{2\rho v}{W/S}\,\frac{C_D(C_D)_v + C_{L_{max}}\left(C_{L_{max}}\right)_v}{\left(C_D{}^2 + C_{L_{max}}{}^2\right)^{0.5}} \\ & - \frac{q}{W/S}\,\frac{\left(C_D(C_D)_v + C_{L_{max}}\left(C_{L_{max}}\right)_v\right)^2}{\left(C_D{}^2 + C_{L_{max}}{}^2\right)^{1.5}} \\ & + \frac{q}{W/S}\,\frac{C_D(C_D)_{vv} + C_{L_{max}}\left(C_{L_{max}}\right)_{vv} + (C_D)_v^2 + \left(C_{L_{max}}\right)_v^2}{\left(C_D{}^2 + C_{L_{max}}{}^2\right)^{0.5}}\end{aligned} \tag{C.129}$$

$$(\delta g's)_h = \frac{1}{2}\,\frac{\partial\rho}{\partial h}\,\frac{v^2}{W/S}\left(C_D{}^2 + C_{L_{max}}{}^2\right)^{0.5} + \frac{q}{W/S}\,\frac{C_D(C_D)_h + C_{L_{max}}\left(C_{L_{max}}\right)_h}{\left(C_D{}^2 + C_{L_{max}}{}^2\right)^{0.5}} \tag{C.130}$$

$$(\delta g's)_{hh} = \frac{1}{2}\frac{\partial^2\rho}{\partial h^2}\frac{v^2}{W/S}\left(C_D{}^2 + C_{L_{max}}{}^2\right)^{0.5}$$

$$+ \frac{\partial\rho}{\partial h}\frac{v^2}{W/S}\frac{C_D(C_D)_h + C_{L_{max}}\left(C_{L_{max}}\right)_h}{\left(C_D{}^2 + C_{L_{max}}{}^2\right)^{0.5}}$$

$$- \frac{q}{W/S}\frac{\left(C_D(C_D)_h + C_{L_{max}}\left(C_{L_{max}}\right)_h\right)^2}{\left(C_D{}^2 + C_{L_{max}}{}^2\right)^{1.5}}$$

$$+ \frac{q}{W/S}\frac{C_D(C_D)_{hh} + C_{L_{max}}\left(C_{L_{max}}\right)_{hh} + (C_D)_h^2 + \left(C_{L_{max}}\right)_h^2}{\left(C_D{}^2 + C_{L_{max}}{}^2\right)^{0.5}}$$

(C.131)

$$(\delta g's)_{vh} = \frac{\partial\rho}{\partial h}\frac{v}{W/S}\left(C_D{}^2 + C_{L_{max}}{}^2\right)^{0.5}$$

$$+ \frac{\rho v}{W/S}\frac{C_D(C_D)_h + C_{L_{max}}\left(C_{L_{max}}\right)_h}{\left(C_D{}^2 + C_{L_{max}}{}^2\right)^{0.5}}$$

$$+ \frac{1}{2}\frac{\partial\rho}{\partial h}\frac{v^2}{W/S}\frac{C_D(C_D)_v + C_{L_{max}}\left(C_{L_{max}}\right)_v}{\left(C_D{}^2 + C_{L_{max}}{}^2\right)^{0.5}}$$

$$- \frac{q}{W/S}\frac{\left(C_D(C_D)_v + C_{L_{max}}\left(C_{L_{max}}\right)_v\right)\left(C_D(C_D)_h + C_{L_{max}}\left(C_{L_{max}}\right)_h\right)}{\left(C_D{}^2 + C_{L_{max}}{}^2\right)^{1.5}}$$

$$+ \frac{q}{W/S} \frac{C_D (C_D)_{vh} + C_{L_{max}} \left(C_{L_{max}}\right)_{vh} + (C_D)_v (C_D)_h + \left(C_{L_{max}}\right)_v \left(C_{L_{max}}\right)_h}{\left(C_D{}^2 + C_{L_{max}}{}^2\right)^{0.5}}$$

(C.132)

$$(\delta g's)_\phi = \frac{q}{W/S} \frac{C_{L_{max}} \left(C_{L_{max}}\right)_\phi}{\left(C_D{}^2 + C_{L_{max}}{}^2\right)^{0.5}} \tag{C.133}$$

$$(\delta g's)_{\phi\phi} = \frac{q}{W/S} \frac{\left(C_{L_{max}}\right)_\phi^2 + C_{L_{max}} \left(C_{L_{max}}\right)_{\phi\phi}}{\left(C_D{}^2 + C_{L_{max}}{}^2\right)^{0.5}} - \frac{q}{W/S} \frac{\left(C_{L_{max}} \left(C_{L_{max}}\right)_\phi\right)^2}{\left(C_D{}^2 + C_{L_{max}}{}^2\right)^{1.5}} \tag{C.134}$$

$$(\delta g's)_\alpha = \frac{q}{W/S} \frac{C_D (C_D)_\alpha + C_{L_{max}} \left(C_{L_{max}}\right)_\alpha}{\left(C_D{}^2 + C_{L_{max}}{}^2\right)^{0.5}} \tag{C.135}$$

$$(\delta g's)_{\alpha\alpha} = \frac{q}{W/S} \frac{(C_D)_\alpha^2 + \left(C_{L_{max}}\right)_\alpha^2 + C_D (C_D)_{\alpha\alpha} + C_{L_{max}} \left(C_{L_{max}}\right)_{\alpha\alpha}}{\left(C_D{}^2 + C_{L_{max}}{}^2\right)^{0.5}}$$

$$- \frac{q}{W/S} \frac{\left(C_D (C_D)_\alpha + C_{L_{max}} \left(C_{L_{max}}\right)_\alpha\right)^2}{\left(C_D^{\,2} + C_{L_{max}}^{\,2}\right)^{1.5}} \tag{C.136}$$

$$(\delta g's)_{\phi\alpha} = \frac{q}{W/S} \frac{\left(C_{L_{max}}\right)_\phi \left(C_{L_{max}}\right)_\alpha + C_{L_{max}} \left(C_{L_{max}}\right)_{\phi\alpha}}{\left(C_D^{\,2} + C_{L_{max}}^{\,2}\right)^{0.5}} - \frac{q}{W/S} \frac{C_{L_{max}} \left(C_{L_{max}}\right)_\phi \left(C_D (C_D)_\alpha + C_{L_{max}} \left(C_{L_{max}}\right)_\alpha\right)}{\left(C_D^{\,2} + C_{L_{max}}^{\,2}\right)^{1.5}} \tag{C.137}$$

$$(\delta g's)_{v\phi} = \frac{\rho v}{W/S} \frac{C_{L_{max}} \left(C_{L_{max}}\right)_\phi}{\left(C_D^{\,2} + C_{L_{max}}^{\,2}\right)^{0.5}} - \frac{q}{W/S} \frac{C_{L_{max}} \left(C_{L_{max}}\right)_\phi \left(C_D (C_D)_v + C_{L_{max}} \left(C_{L_{max}}\right)_v\right)}{\left(C_D^{\,2} + C_{L_{max}}^{\,2}\right)^{1.5}} + \frac{q}{W/S} \frac{\left(C_{L_{max}}\right)_\phi \left(C_{L_{max}}\right)_v + C_{L_{max}} \left(C_{L_{max}}\right)_{v\phi}}{\left(C_D^{\,2} + C_{L_{max}}^{\,2}\right)^{0.5}} \tag{C.138}$$

$$(\delta g's)_{v\alpha} = \frac{\rho v}{W/S} \frac{C_D (C_D)_\alpha + C_{L_{max}}\left(C_{L_{max}}\right)_\alpha}{\left(C_D{}^2 + C_{L_{max}}{}^2\right)^{0.5}}$$

$$- \frac{q}{W/S} \frac{\left(C_D (C_D)_v + C_{L_{max}}\left(C_{L_{max}}\right)_v\right)\left(C_D (C_D)_\alpha + C_{L_{max}}\left(C_{L_{max}}\right)_\alpha\right)}{\left(C_D{}^2 + C_{L_{max}}{}^2\right)^{1.5}}$$

$$+ \frac{q}{W/S} \frac{(C_D)_\alpha (C_D)_v + \left(C_{L_{max}}\right)_\alpha \left(C_{L_{max}}\right)_v + C_D (C_D)_{v\alpha} + C_{L_{max}}\left(C_{L_{max}}\right)_{v\alpha}}{\left(C_D{}^2 + C_{L_{max}}{}^2\right)^{0.5}}$$

(C.139)

$$(\delta g's)_{h\phi} = \frac{1}{2} \frac{\partial \rho}{\partial h} \frac{v^2}{W/S} \frac{C_{L_{max}}\left(C_{L_{max}}\right)_\phi}{\left(C_D{}^2 + C_{L_{max}}{}^2\right)^{0.5}}$$

$$- \frac{q}{W/S} \frac{C_{L_{max}}\left(C_{L_{max}}\right)_\phi \left(C_D (C_D)_h + C_{L_{max}}\left(C_{L_{max}}\right)_h\right)}{\left(C_D{}^2 + C_{L_{max}}{}^2\right)^{1.5}}$$

$$+ \frac{q}{W/S} \frac{\left(C_{L_{max}}\right)_\phi \left(C_{L_{max}}\right)_h + C_{L_{max}}\left(C_{L_{max}}\right)_{h\phi}}{\left(C_D{}^2 + C_{L_{max}}{}^2\right)^{0.5}}$$

(C.140)

$$(\delta g's)_{h\alpha} = \frac{1}{2} \frac{\partial \rho}{\partial h} \frac{v^2}{W/S} \frac{C_D (C_D)_\alpha + C_{L_{max}}\left(C_{L_{max}}\right)_\alpha}{\left(C_D{}^2 + C_{L_{max}}{}^2\right)^{0.5}}$$

$$- \frac{q}{W/S} \frac{\left(C_D (C_D)_h + C_{L_{max}}\left(C_{L_{max}}\right)_h\right)\left(C_D (C_D)_\alpha + C_{L_{max}}\left(C_{L_{max}}\right)_\alpha\right)}{\left(C_D{}^2 + C_{L_{max}}{}^2\right)^{0.5}}$$

$$+ \frac{q}{W/S} \frac{(C_D)_\alpha (C_D)_h + \left(C_{L_{max}}\right)_\alpha \left(C_{L_{max}}\right)_h + C_D (C_D)_{h\alpha} + C_{L_{max}}\left(C_{L_{max}}\right)_{h\alpha}}{\left(C_D{}^2 + C_{L_{max}}{}^2\right)^{0.5}}$$

(C.141)

Terminal boundary conditions for the reverse differential equations (4.3.1) - (4.3.3) are stated in equations (4.3.17) - (4.3.19), in which

$$F_{\underline{x}} = \begin{bmatrix} W_{x_r} \delta x_r \\ W_v \delta v \\ W_\gamma \delta \gamma \\ W_h \delta h \end{bmatrix}^T_{t_f} \qquad (C.142)$$

and

$$F_{\underline{\underline{xx}}} = \left(\left[F_{\underline{x}}\right]^T\right)_{\underline{x}} = \begin{bmatrix} W_{x_r} & 0 & 0 & 0 \\ 0 & W_v & 0 & 0 \\ 0 & 0 & W_\gamma & 0 \\ 0 & 0 & 0 & W_h \end{bmatrix}_{t_f} \qquad (C.143)$$

where

$$\delta x_r = \begin{cases} x_r - x_r\big]_{\text{desired max at } t_f} & \text{if } x_r > x_r\big]_{\text{desired max at } t_f} \\ x_r - x_r\big]_{\text{desired min at } t_f} & \text{if } x_r < x_r\big]_{\text{desired min at } t_f} \\ 0 & \text{otherwise} \end{cases} \tag{C.144}$$

$$\delta v = \begin{cases} v - v\big]_{\text{desired max at } t_f} & \text{if } v > v\big]_{\text{desired max at } t_f} \\ v - v\big]_{\text{desired min at } t_f} & \text{if } v < v\big]_{\text{desired min at } t_f} \\ 0 & \text{otherwise} \end{cases} \tag{C.145}$$

$$\delta \gamma = \begin{cases} \gamma - \gamma\big]_{\text{desired max at } t_f} & \text{if } \gamma > \gamma\big]_{\text{desired max at } t_f} \\ \gamma - \gamma\big]_{\text{desired min at } t_f} & \text{if } \gamma < \gamma\big]_{\text{desired min at } t_f} \\ 0 & \text{otherwise} \end{cases} \tag{C.146}$$

$$\delta h = \begin{cases} h - h\big]_{\text{desired max at } t_f} & \text{if } h > h\big]_{\text{desired max at } t_f} \\ h - h\big]_{\text{desired min at } t_f} & \text{if } h < h\big]_{\text{desired min at } t_f} \\ 0 & \text{otherwise} \end{cases} \tag{C.147}$$

as given by equations (4.2.12) - (4.2.15). All terms in equations (C.142) --(C.147) are evaluated at the terminal time, t_f.

APPENDIX D

CONTROLLABILITY AND OBSERVABILITY OF THE FORMULATION

To examine the controllability and deterministic observability of the system formulation, the second partial reverse differential Eq. (4.3.3),

$$-\dot{J}_{\underline{x}\underline{x}} = H_{\underline{x}\underline{x}} + \underline{f}_{\underline{x}}^{T} J_{\underline{x}\underline{x}} + J_{\underline{x}\underline{x}} \underline{f}_{\underline{x}}$$

$$- (H_{\underline{u}\underline{x}} + \underline{f}_{u}^{T} J_{\underline{x}\underline{x}})^{T} (H_{\underline{u}\underline{u}})^{-1} (H_{\underline{u}\underline{x}} + \underline{f}_{u}^{T} J_{\underline{x}\underline{x}}) \qquad (D.1)$$

must be converted into the standard form for a matrix Riccati equation:

$$\dot{S} = - SF - F^{T}S + SGB^{-1}G^{T}S - A \qquad (D.2)$$

with an appropriate terminal boundary condition, and where

$$S = J_{\underline{x}\underline{x}} \qquad (D.3)$$

$$F = (\underline{f}_{\underline{x}} - \underline{f}_{u} H_{\underline{u}\underline{u}}^{-1} H_{\underline{u}\underline{x}}) \qquad (D.4)$$

$$GB^{-1} G^{T} = \underline{f}_{u} H_{\underline{u}\underline{u}}^{-1} \underline{f}_{u}^{T} \qquad (D.5)$$

$$A = H_{\underline{x}\underline{x}} - H_{\underline{u}\underline{x}}^{T} H_{\underline{u}\underline{u}}^{-1} H_{\underline{u}\underline{x}} \qquad (D.6)$$

The corresponding new time-varying linear system is described by the state equation

$$\dot{\underline{x}}_{new} = F(t)\,\underline{x}_{new} + G(t)\,\underline{u}_{new} \tag{D. 7}$$

The controllability of the formulation is signified by the positive definiteness of the following integral:

$$\int_{t_o}^{t} \Phi(\tau, t_o)\, GB^{-1} G^T \Phi(\tau, t_o)\, d\tau \tag{D. 8}$$

where $\Phi(\tau, t_o)$ is the new system transition matrix.

Though $GB^{-1}G^T$ is not of maximum rank, the integrand of Eq. (D. 8) may integrate to a non-singular term over a finite amount of time to yield a controllable system.

Similiarly, the observability of the formulation is determined by the positive definiteness of

$$\int_{t_o}^{t} \Phi^T(\tau, t_o)\, H^T A^{-1} H\, \Phi(\tau, t_o)\, d\tau \tag{D. 9}$$

where $H^T A^{-1} H$ is the weighting matrix on the state. This we know to be a diagonal positive definite matrix which yields a positive definite integral in Eq. (D. 9), thus verifying the complete observability in the deterministic sense of the system formulation.

APPENDIX E

MAC - 360

MAC is a pseudo-language designed and implemented at the Draper Laboratory of the Massachusetts Institute of Technology to simplify the task of programmer communication with the computer of the mathematics of space mechanics[U-2]. MAC combines the English language with standard algebraic notation into a form which approximates the scientist's or engineer's own language.

The basic MAC feature is a three-card-per-line format, corresponding to the three-levels-per-line of ordinary algebraic notation. For example, the expression

$$a_k = b_k^2 \, c^{de}$$

would appear in MAC as:

```
E                                  2   D E
M                         A   = B  C
S                           K      K
```

Further, the incorporation of a bar, "-", on the E-line over a variable name denotes a vector, while a star, "*", over a variable would define a matrix, both being of an appropriate dimension as defined at the beginning of each program.

Another feature of MAC is the DIFEQ statement, which enables an integration operation in the numerical solution of differential equations. An approximate solution to these is accomplished by a four step Runge-Kutta routine. Logical adjacency of variables separated by a space or some other appropriate delimiter indicates a multiplication, while addition, subtraction, and division are simply denoted by +, -, and /.

This brief introduction to MAC should enable one to follow the logic of the programs as given in Appendix F. For a more detailed explanation, refer to the "Users Guide to MAC - 360" as cited in the bibliography.

APPENDIX F

OPTIMIZATION ALGORITHM COMPUTER PROGRAM LISTINGS

```
//OPTIM DP JOB 4354,CHIN.P,REGION=220K,TIME=1


//OPTIM DP EXEC MACOMPIL


//SYSIN DD *

* PRODM C* PWC1294.OPTIMDDP  SAVESYMB
R
R                 ***************************************************************************************************
R                 MAIN OPTIMIZATION PROGRAM:
R                 THE MAIN DIFFERENTIAL DYNAMIC PROGRAMMING OPTIMIZATION ALGORITHM AND STEP SIZE ADJUSTMENT METHOD ARE
R                 CONTAINED HEREIN. THESE REQUIRE 4 SUBROUTINES:  (1) TO STORE THE INITIAL ESTIMATE OF CONTROL, (2) TO
R                 PROVIDE AN ATMOSPHERE MODEL, (3) TO PROVIDE A VEHICLE AERODYNAMIC CHARACTERISTICS MODEL, AND (4) TO
R                 MINIMIZE THE HAMILTONIAN IN THE CONTROL SPACE TO DETERMINE AN OPTIMAL CONTROL.  PLOTTING AND FILE
R                 STORING FACILITIES ARE ALSO INCLUDED.
R                 THERE ARE 8 INPUT CARDS WHICH ARE DESCRIBED BELOW IN THE PROGRAM REMARKS.
R                 THE OUTPUT INCLUDES AN OPTIMAL TRAJECTORY AND CONTROL, WITH THE CORRESPONDING INFLUENCE FUNCTIONS AND
R                 THE OPTIMAL LINEAR FEEDBACK CONTROLLER.
R                 ***************************************************************************************************

M                 INDEX  R, I, K

M                 DIMENSION  (X,4), (XBAR,4), (DELX,4), (J ,4), (J  ,4X4),
S                                                        X       XX

M                    (FBAR,4), (F,4), (F ,4X4), (F ,4X2), (F   ,4X4), (F  ,4X4),
S                                        X          U        XXR          XV

M                    (F    ,4X4), (F  ,4X4), (F    ,4X4), (F      ,4X4),
S                      XGAM          XH         XPHI          XALPHA

M                    (F    ,4X2), (F      ,4X2), (L ,4), (L  ,4X4), (L  ,2X2),
S                      UPHI          UALPHA          X       XX          UU

M                    (L  ,2X4), (H ,4), (H  ,4X4), (H  ,2X2), (H  ,2X4), (B,2X4),
S                      UX          X       XX         UU         UX

M                    (J  ,4), (J   ,4), (BETA,2X4),
S                      X1        TFX

M                      (KF ,4), (KH  ,4)     ,
S                         U        UX

M                      (T ,255), (V  ,4X4)
S                        O          XX

M                 RESERVE          XR     , V     , GAM     , H     ,
S                                    1201   1201      1201   1201

M                      PHI     , PHIS     , ALPHA     , ALPHAS     ,
S                         1201       1201        1201          1201
```

```
M                       L     , QC     , QCDOT    , GS     , Q     , J     ,
S                        1201    1201       1201     1201   1201    1201

M                       JX0     , JX1     , JX2     , JX3     ,
S                          1201      1201      1201      1201

M                       JXX00     , JXX01     , JXX02     , JXX03     ,
S                            1201       1201       1201        1201

M                       JXX11     , JXX12     , JXX13     , JXX22     ,
S                            1201       1201       1201        1201

M                       JXX23     , JXX33     , A     ,
S                            1201       1201   1201

M                       BETA11     , BETA12     , BETA13     , BETA14     ,
S                             1201        1201         1201        1201

M                       BETA21     , BETA22     , BETA23     , BETA24
S                             1201        1201         1201        1201

M         INIT      STARTER = 1

M                   FIBASE = 0

M                   FTTONM = 1/6076

M                   RHO  = .002378
S                      0

M                   RE = 20902900

M                   G = 32.2

M                   PER = 43

M                   SECORDSW = 0

M                   PLTFIBAS = 3000

M                   NUMDVAR = 13                                                   ** EXCLUDES TIME

R                   *****************************************************************

M                   READ  XR , V , GAM , H , DT, DTM
S                           0   0     0   0

M                   PRINT HDG, (XR         ), V , GAM , H , DT, DTM
S                                0            0     0   0

E
M                    XR (NM)       V        GAM        H         DT        DTM
S                     0            0           0        0

M                   XR  = XR   / FTTONM
S                     0     0
```

```
M        GAM  = GAM  DEGTORAD
S           0      0
R        ****************************************************************
M        READ  W      , W  , W , W
S               QCDOT   GS   Q   QC
M        PRINT HDG, W      , W  , W , W
S                   QCDOT   GS   Q   QC
E
M          W          W          W         W
S           QCDOT      GS          Q         QC
R        ****************************************************************
M        READ  W  , W , W   , W
S               XR   V   GAM   H
M        PRINT HDG,  W  , W , W   , W
S                     XR   V   GAM   H
E
M           W          W        W          W
S            XR          V        GAM         H
E                          2
M        W   = W   FTTONM
S         XR    XR
R        ****************************************************************
M        READ  QCDOT   , GS   , Q   , QC   , XR    , XR
S                   MAX    MAX   MAX    MAX    FMIN    FMAX
M        PRINT HDG, QCDOT   , GS   , Q   , QC   , XR    , XR
S                        MAX    MAX   MAX    MAX    FMIN    FMAX
E
M         QCDOT      GS         Q         QC        XR        XR
S              MAX     MAX        MAX        MAX       FMIN      FMAX
M        XR     = XR     / FTTONM
S          FMIN     FMIN
M        XR     = XR     / FTTONM
S          FMAX     FMAX
R        ****************************************************************
M        READ  V    , V    , GAM    , GAM    , H    , H
S               FMIN   FMAX    FMIN    FMAX   FMIN   FMAX
M        PRINT HDG,  V    , V    , GAM    , GAM    , H    , H
S                     FMIN   FMAX    FMIN    FMAX   FMIN   FMAX
```

```
E
M      V          V         GAM        GAM        H          H
S       FMIN       FMAX       FMIN       FMAX      FMIN       FMAX

M    GAM      = GAM      DEGTORAD
S       FMIN       FMIN

M    GAM      = GAM      DEGTORAD
S       FMAX       FMAX

R    **************************************************************
E                                                                   * PRNTSW=0 PRINTS ALL
M    READ  R , TF, WOS,ETA, ATMODE, PRNTSW                          * PRNTSW=10 PRINTS GOOD FORWARD INTEG'S
S           N                                                       * PRNTSW=11 ==> GOOD FWD'S AND ALL REV'S

M    PRINT HDG, R , TF, WOS,ETA, ATMODE, PRNTSW
S                N

E
M        R          TF        WOS        ETA       ATMODE    PRNTSW
S         N

R    **************************************************************

M    READ PHIH, ALPH, TUSW, PHIT, ALPT, NUMFOBI

M    PRINT HDG, PHIH, ALPH, TUSW, PHIT, ALPT, NUMFOBI

E
M       PHIH       ALPH       TUSW       PHIT       ALPT     NUMFOBI
S

R    **************************************************************
E                                                                   ** REVIPS = NO. REV INTEG'S PER DTM
M    READ PHIDOTM, ALDOTM, ITEREXNO, GF, REVIPS, PLOTCODE           ** ITEREXNO = NO. OF F'WD ITERATIONS
S                                                                              PRIOR TO FILE

E                                                                   ** PLOTCODE NOTEQ 0 ===> PLOTS
M    PRINT HDG,  PHIDOTM, ALDOTM, ITEREXNO, GF, REVIPS, PLOTCODE    ** PLOTCODE=5 PLOTS INITIAL STATES ALSO
S                                                                         (PLOTCODE=5 NOT YET IMPLEMENTED)

E
M    PHIDOT      ALDOT      ITEREXNO     GF       REVIPS    PLOTCODE
S          MAX        MAX

M    PHIDOTM = PHIDOTM DEGTORAD

M    ALDOTM = ALDOTM DEGTORAD

R    **************************************************************

M    IF GF = 0, GF = 1

M    IF REVIPS = 0, REVIPS = 1

R    **************************************************************
```

```
M            DTM = DTM / REVIPS

E                                  2
M            W      = W      RADTODEG   / GF / GF
S             GAM      GAM

R            ***************************************************************

M            SET FILE WRITE (PLTFIBAS - 5)

M            FILE WRITE  PLOTCODE, TF, DT, NUMDVAR

R            ***************************************************************

M            PRINT MSG, SP4

E
M            ***************************************************************
S

M            CALL INITSTOR ,  FIBASE, DT, PER, TF,              ** INCLUDES INITIAL GUESS ON CONTROLS

M                 PHIH, ALPH, TUSW, PHIT, ALPT

M            A  = .0000000001                                   ** A = .0000000001 AT T=0
S             0

M            DO TO STORUNOM FOR K=0(1)(TF/DT)                   ** STORES U
S                                                                          NOMINAL

M            SET FILE READ (FIBASE + 2 K)

M            FILE READ  PHI , ALPHA
S                          K        K

M            PHIS  = PHI
S                K      K

M   STORUNOM ALPHAS  = ALPHA
S                  K        K

M            T1OLD = 0                                          ** TO RESTORE U ON FIRST ITERATIONS ONLY

M   BEGIN    IF STARTER = 0, GO TO INIT

M            OPTIMSW = 0

M            J = 0

M            T = 0                                              ** INITIAL TIME = 0

M            QC = 0

M            QCDOT = 0

M            GS = 0
```

```
M             Q = 0
M             ITERNO = 0
M             NOSTORE = 0
M             K = 0
M    S1FRSAM  DO TO SE
M             ZFSW = 0
M             XR = XR
S                    K
M             V = V
S                  K
M             GAM = GAM
S                      K
M             H = H
S                  K
E             -
M             DELX = ( 0, 0, 0, 0 )
M             CRASHSW = 0
M             IF (ITERNO = 0 OR NOSTORE = 1), GO TO BYPM1
M    RSTRUNM1 DO TO STRUNOM1 FOR K=0(1)((T1-DT)/DT)                       ** RESTORES U          FROM T=0 TO T1-DT
S                                                                                   NOMINAL
M             SET FILE WRITE  (FIBASE + 2 K)
M             FILE WRITE  PHIS , ALPHAS
S                             K        K
M             PHI  = PHIS
S                K        K
M    STRUNOM1 ALPHA  = ALPHAS
S                  K        K
M    BYPM1    IF PRNTSW >=10, IF NOSTORE = 0, GO TO FHD
M             IF PRNTSW NOTEQ 0, GO TO SKPFHD
M    FHD      PRINT FORMAT 1, SP2
M             LONG FORMAT 1
E
M             TIME    XR      VEL    GAMMA   RDOT     ALT     PHI  ALPHA  L
S             (SEC)  (NM)    (FPS)   (DEG)   (FPS)    (FT)   (DEG) (DEG)
```

```
E
M              /D    L/D      QC    QCDOT  G-S     Q      J          JF
S                    MAX                   (FPSS)

M     SKPFHD   K = T/DT

E                                                                            *      -
M              PHI = PHI  DEGTORAD                                      ** U  WITH DELX=(0,0,0,0) INITIALLY
S                       K

M              ALPHA = ALPHA  DEGTORAD
S                           K

M              DO ATMOS

M              DO CLCD

M     SKPPF11  K = T/DT - SGN(T)

M              PP = PHIS  DEGTORAD                                      ** UTILIZED ONCE AT EACH T1
S                       K

M              AA = ALPHAS  DEGTORAD                                    ** PREVIOUS VALUE OF U TO SMOOTH CONTROL
S                         K

M     S1A      K = T/DT

M              PHI = PHI  DEGTORAD                                      * FOR ITERNO = 0 CASE
S                       K

M              ALPHA = ALPHA  DEGTORAD
S                           K

M              IF ITERNO = 0, DO PF1, GO TO S1B

M     UNEWF    K = T/DT

M              PO = PHI    DEGTORAD
S                      K

M              AO = ALPHA     DEGTORAD
S                       K

E              -
M              J  = ( JXO , JX1 , JX2 , JX3  )                           ** RETRIEVAL OF REVERSE INTEGRATION INFO
S               X         K     K     K     K

E              *
M              J   = ( JXX00 , JXX01 , JXX02 , JXX03 ,
S               XX           K       K       K       K

M                      JXX01 , JXX11 , JXX12 , JXX13 ,
S                            K       K       K       K

M                      JXX02 , JXX12 , JXX22 , JXX23 ,
S                            K       K       K       K
```

```
M                  JXX03 , JXX13 , JXX23 , JXX33  )
S                       K       K       K       K
M            IF T >= TF, GO TO S1C
M      S1B   DO TO SSS FOR SCTR=1(1)4
M      S1C   IF ITERNO = 0,  GO TO S1
M            IF T = 0, PHIDTTEM = PHIDOTM, ALDTTEM = ALDOTM,
M                 PHIDOTM = 1800 DEGTORAD, ALDOTM = 1800 DEGTORAD
M            DO MINHU                                                   ** TO OBTAIN U
S                                                                                     NEW
M            IF T = 0, PHIDOTM = PHIDTTEM, ALDOTM = ALDTTEM
M            IF ZFSW = 0,
M            IF SCTR NOTEQ 1, GO TO S1
M            IF PRNTSW >= 10, IF NOSTORE = 0, GO TO PF1
M            IF PRNTSW NOTEQ 0, GO TO BYPPF1
M            GO TO PF1                                                  ** BYPASS EXTRA PRINT
M            IF ITERNO = 0, GO TO PF1
M            IF NOSTORE = 1, GO TO PF1
E                  -
M            PRINT J
S                   X
E                    -    *    -
M            PRINT  ( J  + J   DELX )
S                      X     XX
M            IF (T > 600 OR T < 590), GO TO PF1
E                                  -
M            PRINT  BLANK, BLANK, DELX
M      PF1   PRINT FORMAT 2,T, (XR FTTONM), V, (GAM RADTODEG), (V SIN(GAM)),
M                 H, (PHI RADTODEG), (ALPHA RADTODEG), (C /C ), (L /C ),
S                                                       L  D     L  D
M                                   QC, QCDOT, GS, Q, J, JF
M            LONG FORMAT 2
M            $$$$  $$$$.$  $$,$$$  $$$.$$  $$$.$ $,$$$,$$$  $$$   $$$  $.
M            $$$  $.$$$ $$$,$$$  $$$  $$.$$  $$$ $$$$$.$$$ $$$$$.$$$
```

```
M      BYPPF1   PP = PHI                                                   ** SAVE FOR NEXT U DETERMINATION(SMOOTH)
M               AA = ALPHA
M               IF NOSTORE = 1, GO TO BYPT1
M      STRUT1   DO TO STRUNMT1
M               PHI  = PHI RADTODEG
S                  K
M               ALPHA  = ALPHA RADTODEG
S                    K
M               SET FILE WRITE  (FIBASE + 2 K)
M               FILE WRITE  PHI , ALPHA
S                              K       K
M               PHIS  = PHI RADTODEG                                       ** STORES U         FROM T1 TO TF
S                   K                                                                NOMINAL
M      STRUNMT1 ALPHAS  = ALPHA RADTODEG
S                     K
M      BYPT1    IF ZFSW = 1,  GO TO SE
M      S1       DUMMY = 1
M               DO ATMOS
M               SING = SIN(GAM)
M               COSG = COS(GAM)
M               DO CLCD
E                            2
M               Q = .5 RHO V
M               DXR/DT = V COSG(1/(1 + H/RE))                              ** STATE EQUATIONS
E                                            2
M               DV/DT = -(Q/WOS)G C  + (-G + V /(RE+H))SING
S                                  D
M               DGAM/DT =  (Q/V/WOS)G C  + (V/(RE+H) - G/V)COSG
S                                      L
M               DH/DT = V SING
M               IF V < 0,  V = .000001
E                                     -.5                        3.15
M               QCDOT = .61433466(R )    SQRT(RHO/RHO ) (V/1000)
S                                  N                 O
```

```
E                          2   2
M            GS = (Q/WOS) SQRT(C + L )
S                               D   L

M            IF W        = 0,  GO TO ARGGD
S                QCDOT

M            ARGQCDOT =      (QCDOT - QCDOT    )
S                                        MAX

M            IF ARGQCDOT > 0, DELQCDOT=ARGQCDOT,

M                 OTHERWISE DELQCDOT = 1/SQRT(100 W       )
S                                                  QCDOT

M   ARGGD    IF W    = 0,  GO TO ARGQD
S                GS

M            ARGGS =      (GS - GS    )
S                                MAX

M            IF ARGGS > 0, DELGS=ARGGS, OTHERWISE DELGS = 1/SQRT(100 W   )
S                                                                  GS

M   ARGQD    IF W   = 0,  GO TO LCOMP
S                Q

M            ARGQ =      (Q - Q    )
S                             MAX

M            IF ARGQ > 0, DELQ=ARGQ, OTHERWISE DELQ = 1/SQRT(100 W )
S                                                                Q

E                              2                   2             2
M   LCOMP    L = .5((DELQCDOT) W        + (DELGS) W   + (DELQ) W )          ** QUADRATIC ACCRUED COST TERM
S                                 QCDOT              GS            Q

M            DJ/DT = L

M            DQC/DT = QCDOT

R            ****************************************************************

M   SS       DIFEQ T ,  DT, D XR/DT, D V/DT, D GAM/DT, D H/DT, D J/DT,

M                 D QC/DT

M            IF W    = 0, GO TO SSS
S                QC

M            ARGQC = QC - QC
S                           MAX

M            IF ARGQC > 0, J = J + ARGQC SQRT(W   ), OTHERWISE J = J + .01   ** ADDITIONAL ACCRUED COST TERM FOR QC
S                                            QC

M   SSS      DUMMY = J
```

```
E              -
M              X = ( XR, V, GAM, H )

M              QCDOT = D QC/DT

M              L = D J/DT

M              IF ITERNO = 0,  GO TO CKNSTR

M              K = T/DT

E              -
M              XBAR = ( XR , V , GAM , H  )
S                         K   K     K   K

E              -      -   -
M              DELX = X - XBAR

E              -
M              DELX = ( DELX , DELX , ( GF DELX ), DELX  )
S                           0      1           2       3

M       CKNSTR IF NOSTORE = 1,  GO TO NSTR1

M              K = T/DT

M              XR  = XR                                                ** STORE FORWARD
S                K

M              V  = V
S               K

M              GAM  = GAM
S                 K

M              H  = H
S               K

M              L  = L
S               K

M              QC  = QC
S                K

M              QCDOT  = QCDOT
S                   K

M              GS  = GS
S                K

M              Q  = Q
S               K

M              J  = J
S               K
```

```
M     NSTR1    IF W   = 0, GO TO DELVCALC
S                  XR

M              IF XR > XR     , DELXR = XR - XR     , GO TO DELVCALC
S                        FMAX                 FMAX

M              IF XR < XR     , DELXR = XR - XR     , GO TO DELVCALC
S                        FMIN                 FMIN

M              DELXR = 1/SQRT( 100 W  )
S                                    XR

M     DELVCALC IF W  = 0, GO TO DELGAMCA
S                  V

M              IF V > V     , DELV = V - V     , GO TO DELGAMCA
S                      FMAX             FMAX

M              IF V < V     , DELV = V - V     , GO TO DELGAMCA
S                      FMIN             FMIN

M              DELV = 1/SQRT( 100 W )
S                                  V

M     DELGAMCA IF W    = 0, GO TO DELHCALC
S                  GAM

M              IF GAM > GAM     , DELGAM = GAM - GAM     , GO TO DELHCALC
S                          FMAX                   FMAX

M              IF GAM < GAM     , DELGAM = GAM - GAM     , GO TO DELHCALC
S                          FMIN                   FMIN

M              DELGAM = 1/SQRT( 100 W    )
S                                     GAM

M     DELHCALC DELGAM = DELGAM GF

M              IF W  = 0, GO TO JX1C
S                  H

M              IF H > H     , DELH = H - H     , GO TO JX1C
S                      FMAX             FMAX

M              IF H < H     , DELH = H - H     , GO TO JX1C
S                      FMIN             FMIN

M              DELH = 1/SQRT( 100 W )
S                                  H

E                                       -     -                            -
M     JX1C     IF T = TF - DT, DO JXCALC, J    = J                ** TO OBTAIN J  AT TF-DT
S                                         X1    X                            X

M              JF2 = JF1

M              JF1 = JF
```

```
E                                 2             2                  2                2
M            JF = .5 ( DELXR   W     + DELV   W   + DELGAM   W       + DELH   W   )     ** QUADRATIC TERMINAL COST
S                                XR               V                  GAM              H

M            JMN2 = JMN1

M            JMN1 = JMNO

M            JMNO = J

M    TESTTF  IF T < TF,

M                      GO TO S1A

M                  ZFSW = 1

M            IF ITERNO NOTEQ 0, GO TO UNEWF

M            IF (PRNTSW = 0 OR (PRNTSW >= 10 AND NOSTORE = 0)), DO PF1

M            IF PLOTCODE = 0,  GO TO SE

M            DO TO STRINPLT FOR K=0(1)(TF/DT)                                       ** STORES PTS FOR INITIAL PLOTS

M            SET FILE WRITE  (PLTFIBAS + (NUMDVAR + 1)K)

M    STRINPLT FILE WRITE  (K DT), ALPHA , PHI , GS , Q
S                                      K     K    K   K

M     SE     DUMMY=0

M            JM2 = JMN2 + JF2                                                        ** J AT TF - 2 DT

M            JM1 = JMN1 + JF1                                                        ** J AT TF - DT

M            JMAIN = J + JF                                                          ** TOTAL COST

M            JF9 = JF

M            JMAIN9 = JMAIN

M    UPDATETF DO TO NEWTF

M            A = 0                                                                   ** TERMINAL BOUNDARY CONDITIONS

M    JXCALC  DO TO JXCALCE

E            -
M    JXCALCE J   = ( DELXR W   , DELV W , DELGAM W    , DELH W  )
S             X             XR        V          GAM          H

E            *
M            J   = (  W  ,     0,     0,     0,
S             XX       XR

M                     0,    W ,     0,     0,
S                            V
```

```
M                        0,    0, W    ,     0,
S                               GAM
M                     .  0,    0,    0,    W  )
S                                          H
M           J    = (JMAIN - JM1)/DT
S            TF
E                                              2
M           J      =   (JMAIN - 2 JM1 + JM2)/DT
S            TFTF
E           -        -    -
M           J     = (J   - J   )/DT
S            TFX       X     X1
M           IF J      > 0,
S               TFTF
E                          -     -
M           DELTF = -(J    + J    .DELX)/J     , OTHERWISE
S                      TF     TFX         TFTF
M           DELTF = 0
M           DELTF = DT INTEGER((DELTF)/DT)
M           IF ABS(DELTF) > 5 DT,  DELTF = SGN(DELTF) 5 DT          ** RESTRICTS THE SIZE OF DELTF
M                                            DELTF = 0
M           TFOLD = TF                                               ** TFOLD = OLD TERMINAL TIME
M           TF = TF + DELTF                                          ** UPDATED TERMINAL TIME
M           T = TF                                                   ** INITIALIZES TERMINAL TIME FOR REVERSE
S                                                                                  INTEGRATIONS
M           ITERNO = ITERNO + 1
M           IF ITERNO = NUMFOBI + 1, IF SECORDSW = 0, SECORDSW = 1,  ** NUMFOBI = NO. FIRST ORDER BACKWARD
S                                                                                  INTEGRATIONS
M                OTHERWISE GO TO PRT1R
M           PRINT MSG
M
M           PRINT MSG, SP2
E           *********************
M           SECORDSW SET = 1 HERE
S           *********************
M   PRT1R   IF ITERNO <=1, GO TO NEWTF
```

```
M               PRINT HDG, TF, TFOLD, ITERNO, T1, R, SP2

E
M                 NEWTF     TFOLD     ITERNO      T1        R

M               GO TO SUMMARY

M       NEWTF   PRINT HDG, TF, TFOLD, ITERNO, SP2

E
M                 NEWTF     TFOLD     ITERNO

M       SUMMARY DO TO SUMMARYE

E                          (TF/DT)
M               MAXQCDOT = MAXIMUM ( QCDOT   )
S                            K=0          K

E                       (TF/DT)
M               MAXGS = MAXIMUM ( GS   )
S                         K=0       K

E                      (TF/DT)
M               MAXQ = MAXIMUM ( Q   )
S                        K=0      K

M               K = TF/DT

M               IF XR   > XR     , DELXRP = XR   - XR
S                    K     FMAX              K     FMAX

M               IF XR   < XR     , DELXRP = XR   - XR
S                    K     FMIN              K     FMIN

M               IF (XR  <= XR     AND XR  >= XR     ), DELXRP = 0.0000
S                     K      FMAX       K      FMIN

M               PRINT FORMAT 32,  QC , MAXQCDOT, MAXGS, MAXQ,
S                                  K

M                    ( DELXRP     FTTONM), J , JF, JMAIN, SP4
S                                             K

M               LONG FORMAT 32

E
M                  QC    MAXQCDOT  MAXGS   MAXQ   MISS       J          JF
S               $$$,$$$    $$$     $$.$$   $$$   $$$.$$   $$$$$$.$$$  $$$$$$

E
M                       JMAIN
S               .$$$  $$$$$$.$$$

M       SUMMARYE DUMMY = QC

M               PRINT FORMAT 31
```

```
M             LONG FORMAT 31

E
M             ************************************************************
S

E
M             ************************************************************
S

M             IF ITERNO = ITEREXNO + 1,  EXIT

M             TEFFSW = 0

M             TEFF = 0

M             TB = 0

M             BLOWUPSW = 0

M             T1ZC = 1969

M             DTMM = DTM

M             T77 = 1969

M             DJOA = 0

M             ISW = 0

M             ZRSW = 0

M             C1 = 1.4

M             IF PRNTSW >= 11, GO TO RHD

M             IF PRNTSW NOTEQ 0, GO TO CS1A

M       RHD   PRINT FORMAT 22, SP2

M             LONG FORMAT 22

E                                *      *
M             TIME     A        PHI ALPHA   J        J        J        J
S             (SEC)            (DEG)(DEG)    X,1      X,2      X,3      X,4

E
M                 J       J       J       J       J       J        J
S                  XX,0    XX,1    XX,2    XX,3    XX,5    XX,10    XX,15

M       CS1A  T7 = DT INTEGER((T +.0001)/DT)                      ** T7 = MULTIPLE OF DT

M                                GO TO CS1                         * BYPASS VARIABLE DTMM

M             IF T > T77, GO TO CS1

M             K = T7/DT
```

```
M               IF (T >= (TF-DT) OR T <= DT), GO TO CS1
M               IF ( (PHIS  NOTEQ PHIS     AND T7 NOTEQ T) OR
S                         K            K+1
M                    (PHIS  NOTEQ PHIS     AND T7 = T) ),
S                         K            K-1
M                    DTMM = -.25, OTHERWISE DTMM = DTM, GO TO CS1
M               IF T7 = T, T77 = T7 - DT, OTHERWISE T77 = T7
M       CS1     DUMMY = 0
M               T7 = DT INTEGER((T +.0001)/DT)                              ** T7 = MULTIPLE OF DT
M               K = T7/DT
M               FK = 0
M               V = V  + FK(V     - V  )
S                    K       K+1    K
M               H = H  + FK(H     - H  )
S                    K       K+1    K
E               -
M       ZDELX   DELX = ( 0, 0, 0, 0 )
M               PO = PHIS    DEGTORAD                                       ** HERE, O DENOTES BAR VALUES
S                        K
M               AO = ALPHAS    DEGTORAD
S                          K
M               IF T7 NOTEQ T, GO TO CS2
M               K = T7/DT
M               IF T7 = TF, K = T7/DT - 1
M               PP = PHI     DEGTORAD                                       ** PREVIOUS VALUE OF U TO SMOOTH CONTROL
S                       K+1
M               AA = ALPHA     DEGTORAD
S                         K+1
M               IF ZRSW = 1,  GO TO GETUSTAR
M       CS2     DO TO CS FOR CSCTR=1(1)4
M               IF T = TF,
M                    PHIDTTEM = PHIDOTM, ALDTTEM = ALDOTM,
M                    PHIDOTM = 1800 DEGTORAD, ALDOTM = 1800 DEGTORAD
```

```
E                                                                                *
M     GETUSTAR DO MINHU                                              ** TO OBTAIN U
M              IF T = TF,
M                          PHIDOTM = PHIDTTEM, ALDOTM = ALDTTEM
M              IF ZRSW = 0,
M              IF CSCTR NOTEQ 1, GO TO PBZ
M              IF PRNTSW >= 11, GO TO PB
M              IF PRNTSW NOTEQ 0, GO TO PBWO
M              GO TO PB                                              ** BYPASS EXTRA PRINT
M     PBX      IF ITERNO <= 0, GO TO PB
M              IF PHI = PHIBAR, GO TO PB
E                                                       -    *   -
M              PRINT  (PHIBAR RADTODEG), (PHI RADTODEG), (J  + J   DELX)
S                                                         X    XX
E                                                                  -
M     PB       PRINT FORMAT 21, T, A, (PHI RADTODEG), (ALPHA RADTODEG), J ,
S                                                                       X
E
M                   J     , J    , J    , J    , J    , J     , J
S                    XX,0   XX,1   XX,2   XX,3   XX,5   XX,10   XX,15
M              LONG FORMAT 21
M              $$$$.$$ $$$$.$$$  $$$  $$$  $$$.$$$  $$$.$$$  $$$.$$$  $$$.$
M              $$  $$E $$  $$E $$  $$E $$  $$E $$  $$E $$  $$E $$  $$E $$
M     PBWO     T7 = DT INTEGER((T +.0001)/DT)                        * T7 = MULTIPLE OF DT
M              IF T NOTEQ T7, GO TO PBZ
M     PBW1     K = T7/DT
M     PBW2     DO TO PBZ
E                                                                            *
M              PHI  = PHI RADTODEG                                   ** STORE U
S                 K
M              ALPHA  = ALPHA RADTODEG
S                   K
M              A  = A
S               K
```

```
M        JX0   = J
S           K     X,0
M        JX1   = J
S           K     X,1
M        JX2   = J
S           K     X,2
M        JX3   = J
S           K     X,3
M        IF BLOWUPSW NOTEQ 0, GO TO PBZ
M        JXX00  = J
S             K     XX,0
M        JXX01  = J
S             K     XX,1
M        JXX02  = J
S             K     XX,2
M        JXX03  = J
S             K     XX,3
M        JXX11  = J
S             K     XX,5
M        JXX12  = J
S             K     XX,6
M        JXX13  = J
S             K     XX,7
M        JXX22  = J
S             K     XX,10
M        JXX23  = J
S             K     XX,11
M        JXX33  = J
S             K     XX,15
E                      *
M        IF T = TF, BETA = ( 0,0,0,0,  0,0,0,0 )
M        BETA11  = BETA
S               K        0
M        BETA12  = BETA
S               K        1
M        BETA13  = BETA
S               K        2
```

```
M                BETA14  = BETA
S                      K        3
M                BETA21  = BETA
S                      K        4
M                BETA22  = BETA
S                      K        5
M                BETA23  = BETA
S                      K        6
M                BETA24  = BETA
S                      K        7
M       PBZ      DUMMY = 0
M                IF ZRSW = 1,  GO TO 103
M                DO TO CS FOR RICTR=1(1)REVIPS
M                IF CSCTR NOTEQ 1, GO TO PSSTR
M                T7 = DT INTEGER((T +.0001)/DT)                  ** T7 = MULTIPLE OF DT
M                K = T7/DT
M                FK = 0
M                XR = XR  + FK(XR     - XR )                     ** RETRIEVAL OF FORWARD INTEGRATION INFO
S                       K         K+1     K
M                V = V  + FK(V    - V )
S                     K        K+1    K
M                GAM = GAM  + FK(GAM     - GAM )
S                         K          K+1       K
M                H = H  + FK(H    - H )
S                     K        K+1    K
M                L = L  + FK(L    - L )
S                     K        K+1    K
M                QC = QC  + FK(QC     - QC )
S                       K         K+1     K
M                QCDOT = QCDOT  + FK(QCDOT     - QCDOT )
S                             K           K+1         K
M                GS = GS  + FK(GS     - GS )
S                       K         K+1     K
M                Q = Q  + FK(Q    - Q )
S                     K        K+1    K
```

```
M            PHIBAR = PHIS    DEGTORAD + FK(PHIS     - PHIS ) DEGTORAD       ** BAR DENOTES NOMINAL
S                         K                   K+1        K
M            ALPHABAR = ALPHAS    DEGTORAD + FK(ALPHAS    - ALPHAS )DEGTORAD
S                             K                      K+1         K
M    PSSTR   DO ATMOS
M            SING = SIN(GAM)
M            COSG = COS(GAM)
M            PHITEM = PHI
M            ALPHATEM = ALPHA
M            PHI = PHIBAR, ALPHA = ALPHABAR                                   ** U
S                                                                                NOMINAL
M            DO CLCD
E            -                                            2
M            FBAR = (V COSG/(1+H/RE), (-Q/WOS)G C  +(-G + V /(RE+H))SING,     ** F( U        )
S                                                D                                  NOMINAL
M               (Q/V/WOS)G C  + (V/(RE+H) -G/V)COSG, V SING)
S                          L
M            C      = C , C      = C  , L      = L
S             DBAR     D   LBAR     L     LBAR    L
E                        -  -
M            HBAR = L + J .FBAR                                               ** H( U        )
S                         X                                                         NOMINAL
M            LBAR = L
E                                                                                  *
M            PHI = PHITEM                                                     ** U
M            ALPHA = ALPHATEM
M            DO CLCD
E            -                                            2                        *
M            F    = (V COSG/(1+H/RE), (-Q/WOS)G C  +(-G + V /(RE+H))SING,     ** F( U )
S                                                D
M               (Q/V/WOS)G C  + (V/(RE+H) - G/V)COSG, V SING)
S                          L
M            IF W   = 0,  GO TO LDELCOMP
S               GS
E                                     2       2
M            GSBAR =  (Q/WOS)SQRT(C       + L     )
S                                  DBAR    LBAR
```

```
M           ARGGSBAR =     (GSBAR - GS    )
S                                     MAX

M           IF ARGGSBAR > 0, DELGSBAR = ARGGSBAR,

M               OTHERWISE DELGSBAR = 1/SQRT(100 W  )
S                                                GS

E                                 2    2
M           GS = (Q/WDS)SQRT(C   + L )
S                                 D    L

M           ARGGS =     (GS - GS    )
S                                MAX

M           IF ARGGS > 0, DELGS = ARGGS, OTHERWISE DELGS = 1/SQRT(100 W  )
S                                                                     GS

E                                  2                2
M     LDELCOMP L = L -.5(DELGSBAR) W    + .5(DELGS) W
S                                   GS               GS

E                         - -                                                  *
M           HSTAR = L + J .F                                            ** H( U )
S                         X

R     ***********************************************************************
E                         2    2
M           C = SQRT(C   + L )
S                         D    L

M           C  = C  C    + L  L
S            V    D  DV     L  LV

E                                              2      2
M           C   = C  C     + L  L      + C    + L
S            VV    D  DVV     L  LVV      DV     LV

M           C  = C  C    + L  L
S            H    D  DH     L  LH

E                                              2      2
M           C   = C  C     + L  L      + C    + L
S            HH    D  DHH     L  LHH      DH     LH

M           C    = L  L
S            PHI    L  LPHI

M           C       = C  C         + L  L
S            ALPHA     D  DALPHA      L  LALPHA

E                          2
M           C        = L       + L  L
S            PHIPHI     LPHI      L  LPHIPHI

M           RM = RE + H
```

```
M           RMSQ = RM RM

M           RMCU = RMSQ RM

R     ************************************************************************
E           *
M           F  = ( 0, COSG/(1+H/RE), -V SING/(1+H/RE), -V COSG RE/RMSQ ,
S            X

M                0, (-RHO V/WOS)G C  -(Q/WOS)G C   + (2 V/(RE+H))SING,
S                                 D             DV

E                       2                         2
M                (-G + V /(RE+H))COSG, (-.5 DPDH V /WOS)G C -(Q/WOS)G C  -
S                                                          D            DH

E                       2
M                     V  SING/RMSQ,

E                                                                   2
M                0, (.5 RHO/WOS)G C  + (Q/V/WOS)G C  +(1/(RE+H)+G/V )COSG,
S                                 L                LV

M                (-V/(RE+H)+G/V)SING, (.5 DPDH V/WOS)G C +(Q/V/WOS)G C  -
S                                                      L               LH

M                    V COSG/RMSQ ,

M                  0 ,  SING ,   V COSG ,   0  )

E           -      -
M           F    = F   /GF
S            X,8    X,8

M           F    = F   /GF
S            X,2    X,2

M           F    = F   /GF
S            X,6    X,6

M           F     = F    /GF
S            X,14    X,14

M           F     = F    /GF
S            X,10    X,10

E           *
M           F  =  ( 0, 0,     0, (-Q/WOS)G C        ,
S            U                             DALPHA

M                 (Q/V/WOS)G C     ,(Q/V/WOS)G C       ,      0, 0 )
S                              LPHI              LALPHA

R     ************************************************************************
E           *
M           F    = (0,0,0,0,   0,0,0,0,   0,0,0,0,   0,0,0,0)
S            XXR
```

```
E             *
M             F    = ( 0, 0, -SING/(1+H/RE), -COSG RE/RMSQ,
S              XV

M                    0, (-RHO/WOS)G C -2(RHO V/WOS)G C  -(Q/WOS)G C    +
S                                    D                DV            DVV

M                    (2/(RE+H))SING, (2 V/(RE+H))COSG,  -(DPDH V/WOS)G C  -
S                                                                      D

E                                   2
M                    (.5 DPDH V /WOS)G C  -(RHO V/WOS)G C  -(Q/WOS)G C    -
S                                       DV                DH           DVH

M                       2 V SING/RMSQ ,

E                                                                     3
M                    0 , (RHO/WOS)G C   + (Q/V/WOS)G C    - (2 G/V )COSG,
S                                    LV               LVV

E                                       2
M                    (-1/(RE+H)-G/V )SING, (.5 DPDH/WOS)G C +
S                                                          L

M                    (.5 DPDH V/WOS)G C  +(.5 RHO/WOS)G C  +(Q/V/WOS)G C    -
S                                      LV                LH              LVH

M                       COSG/RMSQ ,

M                    0 ,   0,    COSG ,    0 )

E             -         -
M             F      = F     /GF
S              XV,8     XV,8

M             F      = F     /GF
S              XV,2     XV,2

M             F      = F     /GF
S              XV,6     XV,6

M             F       = F      /GF
S              XV,14     XV,14

M             F       = F      /GF
S              XV,10     XV,10

E             *
M             F      = ( 0, -SING/(1+H/RE), -V COSG/(1+H/RE),V SING RE/RMSQ,
S              XGAM

E                                              2                2
M                      0 , (2 V/(RE+H))COSG, (G-V /(RE+H))SING,-V  COSG/RMSQ,

E                                         2
M                      0 , (-1/(RE+H)-G/V )SING, (-V/(RE+H)+G/V)COSG,
```

```
M                       V SING/RMSQ ,

M                      O,   COSG ,   -V SING ,   0  )

E          -         -
M          F        = F       /GF
S           XGAM,8    XGAM,8

M          F        = F       /GF
S           XGAM,2    XGAM,2

M          F        = F       /GF
S           XGAM,6    XGAM,6

M          F         = F       /GF
S           XGAM,14    XGAM,14

M          F         = F       /GF
S           XGAM,10    XGAM,10

E          *       *
M          F     = F    /GF
S           XGAM    XGAM

E          *
M          F   = ( O, -COSG RE/RMSQ, V SING RE/RMSQ, 2 V COSG RE/RMCU,
S           XH

M                O, (-DPDH V/WOS)G C  -(RHO V/WOS)G C  -
S                                   D                DH

E                              2
M                (.5 DPDH V /WOS)G C   -(Q/WOS)G C    - 2 V SING/RMSQ,
S                                   DV            DVH

E            2
M           -V  COSG/RMSQ,

E                              2                    2
M                  (-.5 D2PDH2 V /WOS)G C  -(DPDH V /WOS)G C   -
S                                        D                 DH

E                                   2
M                (Q/WOS)G C    + 2 V  SING/RMCU,
S                          DHH

M                O, (.5 DPDH/WOS)G C +(.5 RHO/WOS)G C   +
S                                   L                LH

M                (.5 DPDH V/WOS)G C   +(Q/V/WOS)G C   -
S                                  LV              LVH

M                  COSG/RMSQ, V SING/RMSQ,

M                (.5 D2PDH2 V/WOS)G C +(DPDH V/WOS)G C  +(Q/V/WOS)G C   +
S                                    L                LH             LHH
```

```
M                 2 V COSG/RMCU,

M              0 ,   0 ,  0 ,  0  )

E        -       -
M        F     = F    /GF
S         XH,8    XH,8

M        F     = F    /GF
S         XH,2    XH,2

M        F     = F    /GF
S         XH,6    XH,6

M        F       = F     /GF
S         XH,14    XH,14

M        F       = F     /GF
S         XH,10    XH,10

R    *********************************************************************
E        *
M        F     = ( 0,0,    0,0,   (Q/V/WOS)G C           ,
S         UPHI                                 LPHIPHI

M                (Q/V/WOS)G C            ,        0,0 )
S                           LPHIALPHA

E        *
M        F       = ( 0,0,          0,(-Q/WOS)G C             ,
S         UALPHA                                 DALPHAALPHA

M                (Q/V/WOS)G C          ,(Q/V/WOS)G C            , 0,0)
S                           LPHIALPHA                LALPHAALPHA

R    *********************************************************************
E        *
M        F     = ( 0,0,0,0,       0,0,0,0,
S         XPHI

M               0,(.5 RHO/WOS)G C      + (Q/V/WOS)G C      ,
S                               LPHI                LVPHI

M               0,(.5 DPDH V/WOS)G C     + (Q/V/WOS)G C      ,  0,0,0,0)
S                                  LPHI                LHPHI

E        -         -
M        F       = F       /GF
S         XPHI,8    XPHI,8

E        *
M        F       = ( 0,0,0,0,
S         XALPHA

M                 0,(-RHO V/WOS)G C       -(Q/WOS)G C
S                                 DALPHA           DVALPHA
```

```
E                              2
M              0,(-.5 DPDH V /WOS)G C        -(Q/WOS)G C        ,
S                                    DALPHA            DHALPHA

M              0,(.5 RHO/WOS)G C       +(Q/V/WOS)G C       ,
S                              LALPHA              LVALPHA

M              0, (.5 DPDH V/WOS)G C       +(Q/V/WOS)G C       ,
S                                  LALPHA              LHALPHA

M              0, 0, 0, 0 )

E        -            -
M        F          = F          /GF
S         XALPHA,8     XALPHA,8

R    *******************************************************************

M        DELQCDOT   = QCDOT 3.15/V
S                V

M        DELQCDOT    = DELQCDOT   2.15/V
S                VV             V

M        DELQCDOT   = QCDOT .5 DPDH/RHO
S                H

E                                                 2    2
M        DELQCDOT    = QCDOT (.5 D2PDH2/RHO - .25 DPDH /RHO )
S                HH

M        DELQCDOT    = DELQCDOT   3.15/V
S                VH             H

M        DELQCDT    = DELQCDOT
S               V             V

M        DELQCDT    = DELQCDOT
S               VV            VV

M        DELQCDT    = DELQCDOT
S               H             H

M        DELQCDT    = DELQCDOT
S               HH            HH

M        DELQCDT    = DELQCDOT
S               VH            VH

M        IF QCDOT <= QCDOT    ,
S                        MAX

M             DELQCDOT    = 0, DELQCDOT    = 0, DELQCDOT   = 0,
S                     V                VV                H

M             DELQCDOT    = 0, DELQCDOT    = 0
S                     HH               VH
```

```
M     IF QC <= QC    ,
S              MAX

M         DELQCDT    = 0, DELQCDT    = 0, DELQCDT    = 0,
S                V               VV              H

M         DELQCDT    = 0, DELQCDT    = 0
S                HH              VH

R     ****************************************************************

M     DELGS  = (RHO V/WOS)C + (Q/WOS)C /C
S          V                         V

E                                                          2  3
M     DELGS   = (RHO/WOS)C + 2(RHO V/WOS)C /C -(Q/WOS)C /C  +
S          VV                             V            V

M              (Q/WOS)C  /C
S                      VV

E                      2
M     DELGS  = (.5 DPDH V /WOS)C + (Q/WOS)C /C
S          H                               H

E                         2                 2
M     DELGS   = (.5 D2PDH2 V /WOS)C + (DPDH V /WOS)C /C -
S          HH                                      H

E                    2  3
M              (Q/WOS)C /C  + (Q/WOS)C  /C
S                       H              HH

M     DELGS   = (DPDH V/WOS)C + (RHO V/WOS)C /C +
S          VH                               H

E                       2                          3
M              (.5 DPDH V /WOS)C /C - (Q/WOS)C  C /C  +
S                                V            V  H

M              (Q/WOS)(C   C     + L  L     + C   C    + L   L   )/C
S                        D   DVH    L  LVH     DV  DH     LV  LH

M     IF GS <= GS    ,
S              MAX

M         DELGS  = 0, DELGS   = 0, DELGS  = 0, DELGS   = 0,
S              V           VV          H            HH

M         DELGS   = 0
S              VH

R     ****************************************************************

M     DELQ  = RHO V
S         V
```

```
M           DELQ   = RHO
S               VV

E                               2
M           DELQ  = .5 DPDH V
S               H

E                                  2
M           DELQ   = .5 D2PDH2 V
S               HH

M           DELQ   = DPDH V
S               VH

M           IF Q <= Q    ,
S                    MAX

M                DELQ  = 0, DELQ   = 0, DELQ  = 0, DELQ   = 0, DELQ   = 0
S                    V          VV          H           HH           VH

R           ***************************************************************

M           DELGS    = (Q/WOS)C    /C
S                PHI             PHI

E                                                         2    3
M           DELGS        = (Q/WOS)C       /C - (Q/WOS) C   /C
S                PHIPHI           PHIPHI                 PHI

M           DELGS      = (Q/WOS)C      /C
S                ALPHA           ALPHA

M           DELGS            = (Q/WOS)(C   C            + L   L               +
S                ALPHAALPHA            D  DALPHAALPHA     L  LALPHAALPHA

E                     2         2                    2       3
M                    C         + L       )/C -(Q/WOS)C      /C
S                     DALPHA     LALPHA              ALPHA

M           DELGS           = (Q/WOS)(L         L      + L  L           )/C -
S                PHIALPHA             LALPHA  LPHI    L  LPHIALPHA

E                                       3
M                    (Q/WOS)C     C      /C
S                           PHI   ALPHA

M           IF GS <= GS    ,
S                      MAX

M                DELGS     = 0, DELGS        = 0, DELGS       = 0,
S                     PHI            PHIPHI            ALPHA

M                DELGS           = 0, DELGS           = 0
S                     ALPHAALPHA           PHIALPHA

R           ***************************************************************
```

```
E                                                    3
M      DELGS     = (RHO V/WOS)C    /C -(Q/WOS)C  C   /C  +
S           VPHI                PHI             V  PHI

M              (Q/WOS)(L      L    + L  L      )/C
S                       LPHI  LV    L  LVPHI

E                                                          3
M      DELGS       = (RHO V/WOS)C      /C -(Q/WOS)C  C      /C  +
S           VALPHA                ALPHA           V  ALPHA

M              (Q/WOS)(C        C    + L        L    + C  C          +
S                       DALPHA  DV    LALPHA  LV     D  DVALPHA

M              L  L        )/C
S               L  LVALPHA

E                          2                            3
M      DELGS     = (.5 DPDH V /WOS)C    /C -(Q/WOS)C C    /C  +
S           HPHI                   PHI             H PHI

M              (Q/WOS)(L      L    + L  L      )/C
S                       LPHI  LH    L  LHPHI

E                            2                                3
M      DELGS       = (.5 DPDH V /WOS)C      /C -(Q/WOS)C  C      /C  +
S           HALPHA                   ALPHA             H  ALPHA

M              (Q/WOS)(C        C    + L        L    +
S                       DALPHA  DH    LALPHA  LH

M              C  C         + L  L        )/C
S               D  DHALPHA    L  LHALPHA

M      IF GS <= GS    ,
S                MAX

M          DELGS     = 0, DELGS        = 0,
S               VPHI          VALPHA

M          DELGS     = 0, DELGS        = 0
S               HPHI          HALPHA

R ************************************************************************

M      L   = DELQCDOT W      DELQCDOT  + DELGS W   DELGS  +
S       X2           QCDOT          V          GS      V

M              DELQ W  DELQ  +
S                   Q      V

M                   W    DELQCDT
S                    QC        V

M      L   = DELQCDOT W      DELQCDOT  + DELGS W   DELGS  +
S       X4           QCDOT          H          GS      H
```

```
M                     DELQ W  DELQ  +
S                           Q     H

M                          W    DELQCDOT
S                           QC          H

R       ************************************************************************

M               L     = DELGS  W   DELGS     + DELGS W   DELGS
S                UX12        V  GS       PHI            GS       VPHI

M               L     = DELGS  W   DELGS     + DELGS W   DELGS
S                UX14        H  GS       PHI            GS       HPHI

M               L     = DELGS  W   DELGS       + DELGS W   DELGS
S                UX22        V  GS       ALPHA            GS       VALPHA

M               L     = DELGS  W   DELGS       + DELGS W   DELGS
S                UX24        H  GS       ALPHA            GS       HALPHA

R       ************************************************************************
E                                      2
M               L     = W  (DELGS    )  + DELGS W   DELGS
S                UU11    GS       PHI              GS       PHIPHI

M               L     = DELGS       W   DELGS     + DELGS W   DELGS
S                UU12        ALPHA  GS       PHI            GS       PHIALPHA

M               L     = L
S                UU21    UU12

E                                         2
M               L     = W   (DELGS       )  + DELGS W   DELGS
S                UU22    GS        ALPHA              GS       ALPHAALPHA

R       ************************************************************************
E                                         2
M               L     = W     (DELQCDOT )  + DELQCDOT W       DELQCDOT    +
S                XX22    QCDOT           V             QCDOT            VV

E                                      2
M                          W  (DELGS )  + DELGS W   DELGS     +
S                           GS       V             GS       VV

E                                    2
M                          W (DELQ )  + DELQ W  DELQ    +
S                           Q      V          Q      VV

M                              W    DELQCDOT
S                               QC          VV

M               L     = DELQCDOT  W       DELQCDOT  +DELQCDOT W       DELQCDOT  +
S                XX24           H  QCDOT           V            QCDOT          VH

M                          DELGS  W   DELGS  + DELGS W   DELGS   +
S                               H  GS      V            GS       VH
```

```
M                  DELQ  W  DELQ   + DELQ W  DELQ    +   .
S                      H  Q      V          Q      VH

M                      W    DELQCDT
S                       QC           VH

M            L      = L
S             XX42    XX24

E                                     2
M            L      = W      (DELQCDT  ) + DELQCDT W       DELQCDT     +
S             XX44    QCDOT            H               QCDOT           HH

E                                 2
M                  W  (DELGS )  + DELGS W   DELGS    +
S                   GS        H             GS       HH

E                               2
M                  W (DELQ ) + DELQ W  DELQ       +
S                   Q      H          Q     HH

M                      W    DELQCDT
S                       QC           HH

R       *********************************************************************
E            -
M            L  = ( 0, L   , 0, L   )
S             X          X2       X4

E            *
M            L   =( 0, L     , 0, L     ,     0, L     , 0, L      )
S             UX         UX12       UX14          UX22       UX24

E            *
M            L   = ( L     ,  L     ,  L     ,  L      )
S             UU       UU11     UU12     UU21     UU22

E            *
M            L   = ( 0,   0,   0,   0,          0, L     ,  0, L      ,
S             XX                                    XX22        XX24

M                    0,   0,   0,   0,          0, L     ,  0, L      )
S                                                   XX42        XX44

R            *************************************************************
E            *     *        -  *        -  *
M            H   = L     + ( J  F     , J  F        )
S             UX    UX        X  XPHI    X  XALPHA

M            IF T = TF, KH       = 0, OTHERWISE
S                         UUOLD

M            KH       = H                                          ** SAVE PREVIOUS H
S              UUOLD    UU,0                                                        UU11
```

```
E                *     *          -  *         -  *
M                H    = L     + ( J   F      , J   F             )
S                 UU     UU          X   UPHI      X   UALPHA

E                -      -       -  *
M                H   = L    + J   F
S                 X      X       X   X

E                *      *          -  *        -  *       -  *          -  *
M                H     = L     + ( J   F     , J   F   , J   F      , J   F   )
S                 XX      XX          X   XXR      X   XV     X   XGAM      X   XH

M                DA/DT = -(HSTAR - HBAR)                                             ** DDP REVERSE DIFFERENTIAL EQUATIONS

E                  -         -      *      -    -
M       CSBY      DJ /DT =-H   - J    ( F - FBAR ) SGN(SECORDSW)                    ** SECORDSW=1 ACTIVATES SECOND ORDER
S                   X        X     XX                                                                            DDP DE'S

E                      15                          35
M                IF MAXIMUM (ABS(J      )) > 10    ,   IF BLOWUPSW = 0,
S                     I=0            XX,I

M                    BLOWUPSW = -1,    TB = DT INTEGER(T/DT) + DT,

M                    OTHERWISE GO TO OVRFLTST

M                IF PRNTSW >= 11, GO TO PTTB

M                IF PRNTSW NOTEQ 0, GO TO OVRFLTST

M       PTTB     PRINT  HDG, TB, SP2

E
M                   TB

E                                35                         35                                         *
M       OVRFLTST IF ABS(H      ) > 10   ,   H      = SGN(H      ) 10               ** TO PREVENT H   OVERFLOW
S                         UU,0               UU,0        UU,0                                           UU

E                                35                         35
M                IF ABS(H      ) > 10   ,   H      = SGN(H      ) 10
S                         UU,1               UU,1        UU,1

E                                35                         35
M                IF ABS(H      ) > 10   ,   H      = SGN(H      ) 10
S                         UU,2               UU,2        UU,2

E                                35                         35
M                IF ABS(H      ) > 10   ,   H      = SGN(H      ) 10
S                         UU,3               UU,3        UU,3

M                DO TO JXXOVRFL FOR I=0(1)15

E                                35                        35                                          -
M       JXXOVRFL IF ABS(J      ) > 10   , J      = SGN(J      ) 10                 ** TO PREVENT J   OVERFLOW
S                         XX,I              XX,I        XX,I                                            XX
```

```
E              *                                                                        *
M        IF DET(H   ) NZ, GO TO BETACALC                              ** TEST FOR SINGULAR H
S                UU                                                                          UU

M        IF PRNTSW >= 11, GO TO PT&&

M        IF PRNTSW NOTEQ 0, GO TO SKP&&

M   PT&& PRINT MSG

M             **

E        -
M  SKP&& KF   = ( F    , F    , F    , F    )
S          U       U,0    U,2    U,4    U,6

E        -
M        KH   = ( H     , H     , H     , H     )
S          UX       UX,0    UX,1    UX,2    UX,3

M        KH   = H
S          UU     UU,0

M        IF KH   = 0, KH   = KH      , DO FUDGEOLD                    ** H      FUDGE OLD
S              UU       UU     UUOLD                                     UU11

E                                  3
M        IF KH   = 0, KH   = 10  , DO FUDGE                           ** H      FUDGE
S              UU       UU                                               UU11

E        *     *
M        V   = J
S         XX     XX

E         *         *      *T *      *    *
M        DV  /DT = -(H    + F  V   + V   F  -
S          XX          XX     X  XX    XX  X

E                -     -     *                 -     -    *
M            (( KH    + KF  V    )/KH   ) ( KH    + KF  V    ))SGN(SECORDSW)
S                 UX      U  XX      UU       UX      U  XX

E
M                SGN(BLOWUPSW + 1)
S

E                                         -        *
M        DIFEQ T , DTMM ,  DA/DT, DJ /DT, DV  /DT
S                                   X        XX

E        *     *
M        J   = V
S         XX     XX

M        GO TO CS
```

```
E                 *        *    -1  *      *T *
M   BETACALC BETA =-(H   )  ( H    + F   J     )                      ** OPTIMAL LINEAR FEEDBACK CONTROLLER
S                      UU       UX     U   XX

E                 *   *
M                 B = BETA SGN(BLOWUPSW + 1)

E                  *        *     *T *     *   *    *T *   *
M                 DJ   /DT=-(H    + F  J    + J   F  - B  H   B') SGN(SECORDSW)
S                    XX        XX    X  XX    XX  X        UU

E
M                           SGN(BLOWUPSW + 1)
S

E                                         -        *
M                 DIFEQ T , DTMM ,  DA/DT, DJ /DT, DJ  /DT
S                                            X        XX

M   CS            DUMMY = 0

M   SKPPB         IF ABS(A) >ETA, IF TEFFSW = 0, TEFFSW = 1, OTHERWISE GO TO 101

M                 TEFF = DT INTEGER(T/DT)

M                 IF PRNTSW >= 11, GO TO PTTEFF

M                 IF PRNTSW NOTEQ 0,  GO TO 101

M   PTTEFF        PRINT HDG, TEFF, SP2

E
M                    TEFF

M   101           IF BLOWUPSW = -1, GO TO 103

M                 IF T > 0,  GO TO CS1A

M                      T = 0

M                      ZRSW = 1

M                 GO TO CS1

M   103           PRINT HDG,  TF, TEFF, TB, SP2

E
M                     TF        TEFF       TB

M                 PRINT FORMAT 31

M   SAM           IF TEFF <= 0, IF TB <= 0,  GO TO OPTIMAL

M                 NOSTORE = 1

M                 CSAM = .5
```

```
M     ZEROR     R = 0

M               IF BLOWUPSW NOTEQ 0,  T     = 2 TB - TEFF, OTHERWISE        ** T     = TB
S                                      0,0                                      0,1

M               T     = -TEFF                                               ** T     = 0
S                0,0                                                            0,1

M               IF TB >= TEFF,  T     = TEFF - 2 DT                         ** T     = TEFF - DT
S                                0,0                                            0,1

M               IF TEFF = 0, T     = -TEFF
S                             0,0

M               T1PREV2 = -DT

M               T1PREV = -DT

M     T1ITER    T       = .5(TEFF + T      )
S                0,R+1               0,R

M               T1 = DT (ROUND( T       /DT))
S                                0,R+1

M               IF ISW = 0,

M               IF T1 <= T1PREV, T1 = T1 + DT                               ** TO INSURE A DIFFERENT T1 EACH TIME

M               IF (T1 = T1PREV OR T1 = T1PREV2), C1 = 100                  ** PREVENTS LOOPING OR REPS IN T1

M               T1PREV2 = T1PREV

M               T1PREV = T1

M     CZ        PRINT HDG, T1, R, SP2

E
M                   T1          R

M     SKPT1&R   T = T1

M               K = T/DT

M               QC = QC
S                     K

M               QCDOT = QCDOT
S                          K

M               GS = GS
S                     K

M               Q = Q
S                    K

M               J = J
S                    K
```

```
M              AT1 = A
S                     K
M              IF AT1 = 0,  AT1 = A                                   ** FOR TB > 0 CASES
M              DO S1FRSAM                                             ** INTEGRATE FORWARD FROM T1 TO TF
M              J1 = J + JF                                            ** NEW TOTAL COST OPTIMIZING OVER(T1,TF)
M              DELJ = JMAIN -J1                                       ** JMAIN=TOTAL COST FR/PREV FORW'D INTEG
M              PRINT HDG,  J1, DELJ, AT1, (DELJ/ABS(AT1)), SP2
E
M                  J1        DELJ       AT1     DELJ/|AT1|
M              PRINT MSG
E
M              **************************************************************
S
M     TESTT1ZC IF DELJ/ABS(AT1) > 0, IF T1 < T1ZC, T1ZC = T1          ** TO SAVE T1=T1ZC FOR CSAM=0 CASE
M              IF DELJ/ABS(AT1) <= CSAM, GO TO RITER
M              IF R = 0, IF DELJ/ABS(AT1) > C1 , GO TO RITER
M              IF ISW = 0,
M              IF DELJ/ABS(AT1) > C1 , TEFF = T1, R=R+1, T     = T1PREV2,
S                                                        O,R
M                   T1PREVP = T1PREV2, ISW = 1,
M                   GO TO T1ITER
M              IF ISW = 1,
M              IF DELJ/ABS(AT1) > C1 , TEFF = T1, R=R+1, T     = T1PREVP,
S                                                        O,R
M                   GO TO T1ITER
M     CZZ      NOSTORE = 0, T = T1
M              K = T/DT
M              QC = QC
S                     K
M              QCDOT = QCDOT
S                          K
M              GS = GS
S                     K
```

```
M              Q = Q
S                   K

M              J = J
S                   K

M              T1OLD = T1                                                  ** FOR FINAL PRINTOUT

M                             GO TO S1FRSAM                                ** GOOD OPTIMIZATION OVER (T1, TF)

M      RITER   IF T1<TEFF-DT+.001, IF TEFF>0, R=R+1, GO TO T1ITER

M              IF CSAM = 0, GO TO HALT

M              CSAM = 0, DO PCSAMZ

M              IF T1ZC = 1969, GO TO HALT  ,

M                   OTHERWISE T1 = T1ZC, GO TO CZZ

M      HALT    PRINT MSG

E              *******************************************************
M              HALT -- NO IMPROVEMENT ON TRAJECTORY ATTAINABLE
S              *******************************************************

M              GO TO PRINTOUT

M      PCSAMZ  PRINT MSG

E              ***********
M              CSAM SET = 0
S              ***********

M      OPTIMAL PRINT MSG

E              ***************************
M              OPTIMAL HAS BEEN DETERMINED
S              ***************************

M              OPTIMSW = 1

M              IF PLOTCODE = 0,  GO TO PRINTOUT

M              DO TO STROPPLT FOR K=0(1)(TF/DT)                            ** STORES PTS FOR OPTIMAL PLOTS

M              SET FILE WRITE  (PLTFIBAS + 5 + (NUMOVAR + 1)K)

M      STROPPLT FILE WRITE  ALPHAS , PHIS , GS , Q , A ,
S                                 K       K    K   K   K

M                   (XR  FTTONM), V , (GAM  RADTODEG), H
S                       K            K      K               K

M              BETABASE = (FILE WRITE)

M              SET FILE WRITE (PLTFIBAS - 1)
```

```
M               FILE WRITE BETABASE
M               DO TO STRBETA FOR K=0(1)(TF/DT)                              ** STORES PTS FOR OPT. LIN FDBK CONTRL'R
M               SET FILE WRITE (BETABASE + 8 K)
M     STRBETA   FILE WRITE  BETA11 , BETA12 , BETA13 , BETA14 ,
S                                 K        K        K        K
M                           BETA21 , BETA22 , BETA23 , BETA24
S                                 K        K        K        K
M     PRINTOUT  PRINT FORMAT 27
M               LONG FORMAT 27
E
M               TIME   XR     VEL   GAMMA   RDOT   ALT   PHI ALPHA L/D    L/
S               (SEC) (NM)   (FPS)  (DEG)   (FPS)  (FT) (DEG)(DEG)        MA
E
M               D     QC  QCDOT G-S   Q     J     J      J      J      J
S               X            (FPSS)              X,1    X,2    X,3    X,4
M               IF ITERNO > 1, T1 = T1OLD, OTHERWISE T1 = TF + DT
M               DO RSTRUNM1
M               DO TO EX FOR K=0(1)(TF/DT)
M     EX        PRINT FORMAT 26,  (K DT + .00001), (XR  FTTONM), V ,
S                                                    K            K
M                    (GAM  RADTODEG),
S                        K
M                    (V  SIN(GAM  )), H ,PHIS ,ALPHAS , BLANK, BLANK,
S                      K         K      K     K       K
M                    QC , QCDOT , GS , Q , J ,
S                      K       K    K   K   K
M                    JX0 , JX1 , JX2 , JX3
S                      K     K     K     K
M               LONG FORMAT 26
M               $$$$ $$$$.$ $$,$$$  $$.$$  $$$.$ $$$,$$$ $$$ $$$  $.$$$  $.$
M               $$ $$$,$$$ $$$ $$.$$ $$$ $$$$$.$  $$.$$  $$.$$  $$.$$  $$.$$
M               JF = JF9
M               JMAIN = JMAIN9
M               DO SUMMARY
```

```
M                 IF PLOTCODE NZ, IF OPTIMSW = 1,  DO PLOT
M                 EXIT
M       FUDGE     IF (PRNTSW=O OR PRNTSW>=11), DO FUDGE1
M       FUDGE1    PRINT MSG
M                        FUDGE
M       FUDGEOLD  IF (PRNTSW=O OR PRNTSW>=11),  DO FUDGOLD1
M       FUDGOLD1  PRINT MSG
M                        FUDGE OLD
M       ATMOS     DO TO ATMOSE
M                 CALL ATMOSPHR , ATMODE, H, CRASHSW
M       ATMOSE    RESUME  RHO, DPDH, D2PDH2, A    , DADH, D2ADH2, CRASHSW
S                                            SND
M       CLCD      DO TO CLCDE
M                 CALL LIFTDRAG ,  PHI, ALPHA, V, A    , DADH, D2ADH2
S                                                  SND
M       CLCDE     RESUME  C , C , C  , C  , C   , C   , C  , C  , C   , C   ,
S                          D   L   DV   LV   DVV   LVV   DH   LH   DHH   LHH
M                         C   , C   , C    , C        , C      , C      ,
S                          DVH   LVH   LPHI   LPHIPHI   DALPHA   LALPHA
M                         C           , C           , C         ,
S                          DALPHAALPHA   LALPHAALPHA   LPHIALPHA
M                         C     , C     , C      , C      ,
S                          LVPHI   LHPHI   DVALPHA   LVALPHA
M                         C      , C      ,
S                          DHALPHA   LHALPHA
M                               L , L  , L   , L  , L   , L   ,
S                                L   LV   LVV   LH   LHH   LVH
M                               L    , L       , L      , L          ,
S                                LPHI   LPHIPHI   LALPHA   LALPHAALPHA
M                               L         , L     , L     ,
S                                LPHIALPHA   LVPHI   LHPHI
M                               L       , L
S                                LVALPHA   LHALPHA
M       MINHU     DO TO MINHUE
```

```
E                                  -   -     *
M          CALL MINHWRTU,   ATMODE,H, V, J , DELX, J  , WOS, GS   , W  ,
S                                        X         XX        MAX   GS

E
M              PP, AA, PHIDOTM, ALDOTM, PO, AO
S

M   MINHUE  RESUME  PHI, ALPHA

M   PLOT    DO TO PLOTE

M           CALL DDPPLOT,  PLOTCODE, PLTFIBAS, TF, DT, NUMDVAR

M   PLOTE   RESUME

M           START AT BEGIN

/*
```

```
//STORU    JOB 0074,CHIN.P,REGION=220K,TIME=1
// EXEC MACOMPIL
//SYSIN DD *
* PRODM C* PWC1294.STORU      SAVESYMB
R
R              ****************************************************************************************************
R              THIS SUBROUTINE STORES THE INITIAL ESTIMATE OF CONTROL.  IT IS GIVEN AS A STEP FUNCTION OF TIME.
R              ****************************************************************************************************
M     BEGIN    SUBROUTINE  FIBASE, DT, PER, TF, PHIH, ALPH, TUSW, PHIT, ALPT  ** TUSW < 0 TRIGGERS PUSHDOWN
M              TUSWM = TUSW - DT
M              IF TUSW POS, GO TO TRANSIT
M              TUSW = -TUSW
M              TUSWM = TUSW - DT
M              DO TO PUSHDNE FOR T=0(DT)(TF-TUSW)                             ** THIS MOVES TERM. PORT. OF NOM U DOWN
M              K = T/DT
M              SET FILE READ  (FIBASE + 2 K)
M              FILE READ  PTR, ATR
M              SET FILE WRITE  (FIBASE + 2 K + 2 TUSW/DT)
M     PUSHDNE  FILE WRITE  PTR, ATR
M              GO TO HYPER
M     TRANSIT  DO TO TRANSITU FOR T=TUSW(DT)TF
M              K = T/DT
M               SET FILE WRITE (FIBASE + 2 K)
M     TRANSITU FILE WRITE  PHIT, ALPT                                          ** PITCH DOWN MANUEVER
M     HYPER    DO TO HYPERU FOR T=0(DT)TUSWM
M              K = T/DT
M               SET FILE WRITE (FIBASE + 2 K)
M     HYPERU   FILE WRITE  PHIH, ALPH                                          ** HYPERSONIC
M              RETURN
M              START AT BEGIN
/*
```

```
//ATMOS N  JOB 4354,CHIN.P,REGION=220K,TIME=1


//ATMOS N  EXEC MACOMPIL


//SYSIN DD *

* PRODM C* PWC1294.ATMOS4N  SAVESYMB
R
R              ****************************************************************************************************
R              THIS SUBROUTINE PROVIDES AN ATMOSPHERE MODEL.  THE AIR DENSITY IS EXPONENTIALLY VARYING, WITH A SCALE
R              HEIGHT OF 28,500 FEET.  THE SPEEDS OF SOUND ARE EXTRACTED FROM THE 1962 ICAO ATMOSPHERE TABLES.  A 4-
R              POINT NEWTON CUBIC FIT INTERPOLATION SCHEME IS UTILIZED.
R              ****************************************************************************************************

M              INDEX  N

M              DIMENSION  (A,16X4), (X,4), (Y,4)

M       INIT   STARTER = 1

M              RHO  = .002378
S                 0

M              H = 28500
S               S

M              A  = 1116
S               0

M              A  = 1097
S               1

M              A  = 1077
S               2

M              A  = 1057
S               3

M              A  = 1037
S               4

M              A  = 1016
S               5

M              A  = 995
S               6

M              A  = 973
S               7

M              A  = 968
S               8
```

M S	$A_{9} = 968$
M S	$A_{10} = 968$
M S	$A_{11} = 968$
M S	$A_{12} = 968$
M S	$A_{13} = 968$
M S	$A_{14} = 971$
M S	$A_{15} = 974$
M S	$A_{16} = 978$
M S	$A_{17} = 981$
M S	$A_{18} = 985$
M S	$A_{19} = 988$
M S	$A_{20} = 991$
M S	$A_{21} = 995$
M S	$A_{22} = 1004$
M S	$A_{23} = 1013$
M S	$A_{24} = 1022$
M S	$A_{25} = 1031$
M S	$A_{26} = 1040$
M S	$A_{27} = 1049$

M S	$A_{28} = 1058$
M S	$A_{29} = 1066$
M S	$A_{30} = 1075$
M S	$A_{31} = 1082$
M S	$A_{32} = 1082$
M S	$A_{33} = 1082$
M S	$A_{34} = 1082$
M S	$A_{35} = 1077$
M S	$A_{36} = 1071$
M S	$A_{37} = 1064$
M S	$A_{38} = 1058$
M S	$A_{39} = 1052$
M S	$A_{40} = 1046$
M S	$A_{41} = 1033$
M S	$A_{42} = 1020$
M S	$A_{43} = 1007$
M S	$A_{44} = 994$
M S	$A_{45} = 981$
M S	$A_{46} = 967$

```
M                A   = 953
S                 47
M                A   = 939
S                 48
M                A   = 925
S                 49
M                A   = 911
S                 50
M                A   = 896
S                 51
M                A   = 884
S                 52
M                A   = 884
S                 53
M                A   = 884
S                 54
M                A   = 884
S                 55
M                A   = 884
S                 56
M                A   = 884
S                 57
M                A   = 884
S                 58
M                A   = 893
S                 59
M                A   = 904
S                 60
M                A   = 904
S                 61
M                A   = 904
S                 62
M                A   = 904
S                 63
M                GO TO B1
M       BEGIN    SUBROUTINE  ATMODE, H, CRASHSW
M                IF STARTER = 0, GO TO INIT
```

```
M     B1     IF H > 0, GO TO B2

M            IF CRASHSW = 1,  GO TO B4

M            PRINT MSG, SP2

E            *********************
M            STS CRASHES INTO EARTH
S            *********************

M            CRASHSW = 1

M     B4     H = 0

M            A = A
S                 0

M            DADH = 0

M            D2ADH2 = 0

M            GO TO B3

M     B2     I = INTEGER(H/5000)

M            IF I > 60,  A = A  , DADH = 0, D2ADH2 = 0, GO TO B3
S                             60

M            F = H - I 5000

M            N = I - 1

M            IF I > 58,   N = 57

M            IF I < 2,   N = 0

E            -
M            X = ( 5000 N, 5000(N+1), 5000(N+2), 5000(N+3)   )

M            U0 = H - X
S                      0

M            U1 = H - X
S                      1

M            U2 = H - X
S                      2

E            -   -
M            Y = A
S                 N

M            P01 = (Y   - Y )/(X   - X )
S                    0     1     0     1

M            P12 = (Y   - Y )/(X   - X )
S                    1     2     1     2
```

```
M              P23 = (Y  - Y )/(X  - X )
S                      2    3    2    3

M              P012 = (P01 - P12)/(X  - X )
S                                     0    2

M              P123 = (P12 - P23)/(X  - X )
S                                     1    3

M              P0123 = (P012 - P123)/(X  - X )
S                                        0    3

M              A = A      + U0 P01 + U0 U1 P012 + U0 U1 U2 P0123
S                   N

M              DADH = P01 + (U0 + U1) P012 + (U0 U1 + U1 U2 + U0 U2) P0123

M              D2ADH2 = 2 P012 + 2 (U0 + U1 + U2) P0123

M       B3     RHO = RHO  EXP(-H/H )
S                       0        S

M              DPDH = -RHO/H
S                           S

E                            2
M              D2PDH2 = RHO/H
S                            S

M              RETURN  RHO, DPDH, D2PDH2, A, DADH, D2ADH2, CRASHSW

M              START AT BEGIN

/*
```

```
//CLCDH 4N JOB 0074,CHIN.P,REGION=220K,TIME=1


//CLCDH 4N EXEC MACOMPIL


//SYSIN DD *

* PRODM C* PWC1294.CLCDHY4N  SAVESYMB
R
R               ***************************************************************************************************
R               THIS SUBROUTINE PROVIDES A VEHICLE AERODYNAMIC CHARACTERISTICS MODEL.  IT GIVES THE COEFFICIENTS OF
R               LIFT AND DRAG ALONG WITH THEIR DERIVATIVES AS A FUNCTION OF A HYPERSONIC MACH NUMBER, ANGLE OF ATTACK,
R               ALPHA, AND ROLL ANGLE, PHI.  A 4-POINT NEWTON CUBIC FIT INTERPOLATION SCHEME IS UTILIZED.
R               ***************************************************************************************************

M               INDEX  N, J, H

M               DIMENSION  (X,4),

M                  (Y,4), (ALPHA,4X4), (C ,4X4), (C ,4X4)
S                                       L         D

M               RESERVE  ALPHA
S                             16

M         INIT  STARTER = 1

M               IF PHI = 1971, LDSF = AL, OTHERWISE LDSF = 1                        ** SETS LDSF

M               IF (PHI = 1971 AND AL < .000001), LDSF = 1

M               L    = 0
S                LM

M               L     = 0
S                LMM

M               L      = 0
S                LMAL

M               C    = 0
S                DM

M               C     = 0
S                DMM
```

```
M              C     = 0
S               DMAL

E              *
M              ALPHA = (   0,2,4,5,7,10,15,20,30,40,50,55,60,70,80,90)
S                   1

E              *
M              C  = ( .142, .145, .148, .15, .166, .191, .283, .443,
S               D

M                     1, 1.88, 2.99, 3.58, 4.16, 5.18, 5.86, 6.05 )

E              *
M              C  = (-.0171, .00894, .035, .0822, .148, .246, .478, .756,
S               L

M                     1.38, 1.94, 2.24, 2.26, 2.17, 1.68, .86, -.141 )

E              *         *
M              C  = LDSF C
S               L         L

M              GO TO P10

M      4NFIT   DO TO 4NFITE

M              P01 = (Y  - Y )/(X  - X )
S                      0    1    0    1

M              P12 = (Y  - Y )/(X  - X )
S                      1    2    1    2

M              P23 = (Y  - Y )/(X  - X )
S                      2    3    2    3

M              P012 = (P01 - P12)/(X  - X )
S                                   0    2

M              P123 = (P12 - P23)/(X  - X )
S                                   1    3

M              P0123 = (P012 - P123)/(X  - X )
S                                      0    3

M      4NFITE  DUMMY = P01

M      BEGIN   SUBROUTINE  PHI, AL, V, A, DADH, D2ADH2

M              IF STARTER = 0, GO TO INIT

M      P10     IF AL < 0,  AL = 0,  OTHERWISE GO TO P3

M              PRINT MSG, SP2
```

```
E           *************************
M           ALPHA < 0 --- SET IT TO 0
S           *************************

M     P3    IF AL = 0,  AL = .000000001

M           IF AL > 90 DEGTORAD, AL = 90 DEGTORAD, OTHERWISE GO TO P4

M           PRINT MSG, SP2

E           ***********************************
M           ALPHA > 90 DEG --- SET IT TO 90 DEG
S           ***********************************

M     P4    IF V <= 0,  V = .000000001

M           M = V/A

M           AL = AL RADTODEG

E                  16
M           H = SATISFY (ALPHA  >= AL ) - 1
S                 J=1          J

M           N = J - 2

M           IF J < 3, N = 1

M           IF J > 15, N = 13

E           -   -
M           X = ALPHA
S                    N

M           U0 = AL - X
S                      0

M           U1 = AL - X
S                      1

M           U2 = AL - X
S                      2

E           -   -
M           Y = C
S                L,N-1

M           DO 4NFIT

M           L   = C       + U0 P01 + U0 U1 P012 + U0 U1 U2 P0123
S            L     L,N-1

M           L     = P01 + (U0 + U1) P012 + (U0 U1 + U1 U2 + U0 U2) P0123
S            LAL

M           L       = 2 P012 + 2 (U0 + U1 + U2) P0123
S            LALAL
```

```
E               -   -
M               Y = C
S                    D,N-1

M               DO 4NFIT

M               C  = C        + U0 P01 + U0 U1 P012 + U0 U1 U2 P0123
S                D    D,N-1

M               C     = P01 + (U0 + U1) P012 + (U0 U1 + U1 U2 + U0 U2) P0123
S                DAL

M               C       = 2 P012 + 2 (U0 + U1 + U2) P0123
S                DALAL

R       ***********************************************************************

M       P16     SINP = SIN(PHI)

M               COSP = COS(PHI)

M               C  = L  COSP
S                L    L

M               C     = -L  SINP
S                LPHI      L

M               C          = -C
S                LPHIPHI      L

M               L     = L     RADTODEG
S                LAL     LAL

M               C     = C     RADTODEG
S                DAL     DAL

E                                        2
M               L       = L       RADTODEG
S                LALAL     LALAL

E                                        2
M               C       = C       RADTODEG
S                DALAL     DALAL

M               C       = -SIN(PHI) L
S                LPHIAL                LAL

M               C     = L     COSP
S                LAL     LAL

M               C       = L       COSP
S                LALAL     LALAL

M               RETURN   C , C , 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
S                         D   L
```

```
M                    C    , C        , C   , C   ,
S                     LPHI   LPHIPHI   DAL   LAL
M                    C     , C     , C        ,
S                     DALAL   LALAL   LPHIAL
M                    0, 0, 0, 0, 0, 0,
S
M                    L , 0, 0, 0, 0, 0, 0, 0, L   , L     , 0, 0, 0, 0, 0
S                     L                        LAL   LALAL
M              START AT BEGIN
/*
```

```
//MINHY 05 JOB 0074,CH V.P,REGION=220K,TIME=1

// EXEC MACOMPIL
//SYSIN DD *
* PRODM C* PWC1294.MINHY605  SAVESYMB
R
R              ****************************************************************************************************
R              THIS SUBROUTINE MINIMIZES THE HAMILTONIAN IN THE CONTROL SPACE TO DETERMINE AN OPTIMAL CONTROL.
R              ****************************************************************************************************
M              INDEX  I, B, K, N, Z
M              DIMENSION  (J      ,4), (J ,4), (DELX,4), (J   ,4X4) ,
S                          XMINU       X                  XX
M                   (BU,1X161),  (RU,1X161),  (ALPHU,1X161),  (PHIMINU,1X255),
M                   (ALMINU,1X255) ,
M                   (CU ,16),  (CU ,16)
S                      D          L
M              DIMENSION  (ALPHA,17)
M       INIT   GS = 32.2
M              STARTER = 1
E              -
M              ALPHA = (0, 0,2,4,5,7,10,15,20,30,40,50,55,60,70,80,90)
E              -
M              CU  =(.142, .145, .148, .15, .166, .191, .283, .443,
S                D
M                   1, 1.88, 2.99, 3.58, 4.16, 5.18, 5.86, 6.05 )
E              -
M              CU  =(-.0171, .00894, .035, .0822, .148, .246, .478, .756,
S                L
M                   1.38, 1.94, 2.24, 2.26, 2.17, 1.68, .86, -.141 )
M              GO TO U3
E                                            -    -     *
M       BEGINU SUBROUTINE   ATMODE, H, V, J , DELX, J  , WOS, GS   , W  , PP,
S                                             X          XX        MAX   GS
M                   AA, PHIDOTM, ALDOTM, PO, AO
M              IF STARTER = 0, GO TO INIT
```

```
E                        60
M     U3      HMINU = 10

M             K = 0

M             Z = 0

M             CALL ATMOSPHR , ATMODE, H, 1

M             RESUME  RHO, DPDH, D2PDH2, A, DADH, D2ADH2, CRASHSW

M             IF V >= 0, GO TO U5

M             V = 0

M             PRINT MSG

E             ********************
M             V < 0 --- V SET TO 0
S             ********************

E                          2
M     U5      Q = .5 RHO V

E             -         -    *   -
M             J       = J  + J    DELX
S              XMINU     X     XX

M               LAM1 = J
S                       XMINU,1

M               LAM2 = J
S                       XMINU,2

M             IF V/A > 5, B=13, GO TO M5

M             DO TO U6 FOR B=1(1)13

M     M5      C    = CU
S              DU      D,B-1

M             C    = CU
S              LU      L,B-1

E                                   2   2
M             ARGGSU = (Q/WOS)SQRT(C   +C   ) - GS
S                                   DU  LU       MAX

M             IF ARGGSU > 0 , DELGSU = ARGGSU, OTHERWISE DELGSU = 0

M             DO TO U6 FOR R=0(180)180

M             COSR = COS(R DEGTORAD)

E                              2
M             HU = .5 W    DELGSU  +.5 RHO V(-LAM1 V C   + LAM2 C    COSR)GS/WOS
S                      GS                             DU         LU
```

```
M             IF HU = HMINU,  K=K+1,  BU =B,
S                                        K
M                RU =R DEGTORAD, ALPHU =ALPHA  DEGTORAD
S                  K                   K      B
M    U6       IF HU < HMINU,  HMINU = HU, K=0, BU =B,
S                                                 K
M                RU =R DEGTORAD, ALPHU = ALPHA  DEGTORAD
S                  K                   K       B
M             HMINUF = HMINU
M             GO TO U8
M             DO TO U8 FOR N=0(1)K
M             B = BU   , R=RU  RADTODEG
S                   N        N
M             IF B=1, UA=ALPHA ,                          OTHERWISE
S                             B
M                     UA=ALPHA
S                             B-1
M             IF B=15, UA1=ALPHA , OTHERWISE
S                               B
M                     UA1=ALPHA
S                              B+1
M             DO TO U8 FOR  C=UA(1)UA1
M             IF C = ALPHA , GO TO U8
S                         B
M             IF B=1, C   =CU     +((C-UA)/(UA1-UA))(CU    -CU      ),
S                      DU   D,B-1                       D,B   D,B-1
M                     C   =(CU     +((C-UA)/(UA1-UA))(CU    -CU      ))
S                      LU     L,B-1                       L,B   L,B-1
M             IF B=15, C   =CU     +((C-UA)/(UA1-UA))(CU      -CU      ),
S                       DU   D,B-2                       D,B-1   D,B-2
M                      C   = (CU     +((C-UA)/(UA1-UA))(CU      -CU      ))
S                       LU     L,B-2                       L,B-1   L,B-2
M             IF B NOTEQ 1, IF B NOTEQ 15, IF C < ALPHA ,
S                                                 B
M                C   =CU     +((C-UA)/(ALPHA -UA))(CU      -CU      ) ,
S                 DU   D,B-2              B        D,B-1   D,B-2
```

```
M                 C   =(CU      +((C-UA)/(ALPHA -UA))(CU      -CU      ))
S                  LU     L,B-2                  B         L,B-1   L,B-2

M             IF B NOTEQ 1, IF B NOTEQ 15, IF C=ALPHA ,
S                                                   B

M                 C   =CU         ,
S                  DU    D,B-1

M                 C   =CU
S                  LU    L,B-1

M             IF B NOTEQ 1, IF B NOTEQ 15, IF C > ALPHA ,
S                                                     B

M                 C   =CU      +((C-ALPHA )/(UA1-ALPHA ))(CU    -CU      ) ,
S                  DU    D,B-1           B              B      D,B   D,B-1

M                 C   =(CU      +((C-ALPHA )/(UA1-ALPHA ))(CU    -CU      ))
S                  LU     L,B-1           B              B      L,B   L,B-1

E                                  2    2
M             ARGGSU = (Q/WOS)SQRT(C    +C  ) - GS
S                                  DU   LU      MAX

M             IF ARGGSU > 0, DELGSU = ARGGSU, OTHERWISE DELGSU = 0

M             Y = R

M             IF Y DEGTORAD = RU , GO TO U8
S                              N

M                       COSY = COS(Y DEGTORAD)

E                              2
M             HU = .5 W   DELGSU  +(Q/WOS) (-LAM1 C    + LAM2 C    COSY/V) GS
S                     GS                           DU         LU

M             IF HU = HMINUF,Z = Z + 1, PHIMINU  = Y DEGTORAD,
S                                              Z

M                 ALMINU  = C DEGTORAD
S                       Z

M       U7    IF HU < HMINUF, HMINUF= HU,  Z = 0,

M                 ALMINU  = C DEGTORAD, PHIMINU  = Y DEGTORAD
S                       Z                      Z

M       U8    DUMMY = C Y

M             IF HMINU = HMINUF,  IF Z = 0,  OTHERWISE GO TO U11            ** TIES FROM 2 REGIONS ARE NOT COMPARED

M             IF K = 0, N = 0,  GO TO U12
```

```
E                 K
M         N = SATISFY (ABS(AO - ALPHU ) +ABS(PO- RU ) MINIMUM )
S                I=0                  I              I

M   U12   PHI1 = RU ,  AL1 = ALPHU
S                  N              N

M         PHIMINU  = RU
S                N     N

M         ALMINU  = ALPHU
S               N        N

M         IF ABS(RU  - PP) > PHIDOTM,
S                  N

M              PHIMINU  =  PP + PHIDOTM SGN(RU  - PP)
S                     N                        N

M         IF ABS(ALPHU  - AA) > ALDOTM,
S                     N

M              ALMINU  = AA + ALDOTM SGN(ALPHU  - AA)
S                    N                         N

M         GO TO U10

M   U11   IF Z=0,  N=0,  GO TO U9

E                 Z
M         N = SATISFY (ABS(AO-ALMINU )+ABS(PO-PHIMINU ) MINIMUM )
S                I=0                 I                 I

M   U9    PHI1 = PHIMINU , AL1 = ALMINU
S                       N               N

M         IF ABS(PHIMINU  - PP) > PHIDOTM,
S                       N

M              PHIMINU  = PP + PHIDOTM SGN(PHIMINU  - PP)
S                     N                            N

M         IF ABS(ALMINU  - AA) > ALDOTM,
S                      N

M              ALMINU  = AA + ALDOTM SGN(ALMINU - AA)
S                    N                          N

M   U10   RETURN  PHIMINU , ALMINU                               ** IN RADIANS
S                       N        N

M         START AT BEGINU

/*
```

References

A-1 "A Two-Stage Fixed Wing Space Transportation System-Final Report", McDonnell Douglas Corporation Report MDC E0056, Vols. I-II, December, 1969.

A-2 Allen, H. J., and A. J. Eggers, Jr., "A Study of the Motion and Aerodynamic Heating of Missles Entering the Earth's Atmosphere at High Supersonic Speeds", NASA Technical Note 4047, Washington, D. C., October, 1957.

A-3 Athans, M., and P. L. Falb, Optimal Control: An Introduction to the Theory and Its Applications, McGraw-Hill, New York, 1966.

B-1 Bryson, A. E., and Y. C. Ho, Applied Optimal Control: Optimization, Estimation, and Control, Blaisdell Company, Waltham, Massachusetts, 1969.

C-1 Chapman, D. R., "An Approximate Analytical Method for Studying Entry into Planetary Atmospheres", NASA Technical Report R-11, Washington, D. C., 1959.

C-2 Chin, P. W., Jr., "Hypersonic Reentry Phase for the Space Transportation System (STS) (formerly ILRV)", 23A STS Memo No. 1-69, M. I. T. Instrumentation Laboratory, Cambridge, Massachusetts, September 11, 1969.

C-3 Chin, P. W., Jr., "Entry Flight Data From Apollo 12", M. I. T. Draper Laboratory Report E-2482, M. I. T. Draper Laboratory, Cambridge, Massachusetts, March, 1970.

D-1 Dreyfus, S. E., Dynamic Programming and the Calculus of Variations, Academic Press, New York, 1965.

F-1 Faget, M. A., "Space Shuttle: A New Configuration", Astronautics and Aeronautics, Vol. 8, No. 1, January, 1970.

F-2 Foudriat, E. C., "Study of the Use of a Terminal Controller Technique for Reentry Guidance of a Capsule-Type Vehicle", NASA Technical Note D-828, Washington, D. C., May, 1961.

F-3 Fraser, D. C., "Development of a Control System for the STS", M. I. T. IL Memo, M. I. T. Instrumentation Laboratory, Cambridge, Massachusetts, October, 1969.

G-1 GSOP, "Guidance System Operations Plan for Manned CM Earth Orbital and Lunar Missions Using Program Colossus 2C (Comanche 67)", Section 5: Guidance Equations (Rev: 8), M. I. T. Instrumentation Laboratory, Cambridge, Massachusetts, July, 1969.

H-1 Hildebrand, F. B., Advanced Calculus for Applications, Prentice-Hall, Englewood Cliffs, New Jersey, 1965.

H-2 Hildebrand, F. B., Methods of Applied Mathematics, Prentice-Hall, Englewood Cliffs, New Jersey, 1965.

H-3 "Hypersonic Aerodynamic Characteristics of the Revised Baseline Orbiter (August 15, 1969)", Aerodynamics and Entry Technology Section Memorandum EX24/6910-251C, Manned Spacecraft Center, Houston, Texas, October 3, 1969.

J-1 Jacobson, D. H., "Differential Dynamic Programming Methods for Determining Optimal Control of Non-Linear Systems", University of London, Ph. D Thesis, October, 1967.

J-2 Jacobson, D. H., "New Second-Order and First-Order Algorithms for Determining Optimal Control: A Differential Dynamic Programming Approach", Journal of Optimization Theory and Applications, Vol. 2, No. 6, November, 1968.

J-3 Jacobson, D. H., and D. Q. Mayne, Differential Dynamic Programming, American Elsevier, New York, 1970.

K-1 Kuo, B. C., Automatic Control Systems, Prentice-Hall, Englewood Cliffs, New Jersey, 1967.

L-1 Lee, R. C. K., "Optimal Estimation, Identification, and Control", Research Monograph No. 28, M. I. T. Press, Cambridge, Massachusetts, 1964.

L-2 Lessing, H. C., P. J. Tunnell, and R. E. Coate, "Lunar Landing and Long-Range Earth Re-Entry Guidance by Application of Perturbation Theory", Journal of Spacecraft, Vol. 1, No. 2, March-April, 1964.

L-3 Levinsky, E. S., "Application of Inequality Constraints to Variational Problems of Lifting Re-Entry", Journal of the Aerospace Sciences, Vol. 29, No. 4, April, 1962.

L-4 Lickly, D. J., H. R. Morth, and B. S. Crawford, "Apollo Reentry Guidance", M. I. T. IL Report R-415, M. I. T. Instrumentation Laboratory, Cambridge, Massachusetts, July, 1963.

M-1 Manned Spacecraft Center (MSC) Shuttle Study Status Report, Houston, Texas, May 21, 1969.

M-2 Manned Spacecraft Center (MSC) ILRV Space Shuttle Status Report, Houston, Texas, August 13, 1969.

M-3 Manned Spacecraft Center (MSC) ILRV Shuttle Status Briefing, Houston, Texas, September 30, 1969.

M-4 Morth, R., "Reentry Guidance for Apollo", M.I.T. IL Report R-532, M.I.T. Instrumentation Laboratory, Cambridge, Massachusetts, January, 1966.

M-5 Morth, R., "Entry Trajectory Reconstruction Using Flight Data", paper presented at IFAC Symposium, Toulouse, France, March, 1970.

N-1 "NASA Space Vehicle Design Criteria (Guidance and Control): Guidance and Navigation for Entry Vehicles", NASA SP-8015, Washington, D.C., November, 1968.

O-1 "Orbiter Transition Using New Aero Data", Applied Flight Dynamics Section Memorandum EX22/6908-175C, Manned Spacecraft Center, Houston, Texas, August 11, 1969.

S-1 Stengel, R.F., "Optimal Transition from Entry to Cruising Flight", M.I.T. Draper Laboratory Report E-2522, M.I.T. Draper Laboratory, Cambridge, Massachusetts, July, 1970.

U-1 U.S. Standard Atmosphere, 1962, NASA, USAF, U.S. Weather Bureau, Washington, D.C., December, 1962.

U-2 "Users Guide to MAC-360", M.I.T. Instrumentation Laboratory, Cambridge, Massachusetts, September, 1967.

W-1 Wingrove, R. C., "A Survey of Atmosphere Reentry Guidance and Control Methods", IAS Paper No. 63-86, January, 1963.